卒業論文

環境・生体データログからの
勾配・制約を考慮したPSOによる
行動パターン解析

Similarity and Event Detection by Behavior Pattern Analysis from Environment Recognition Life Log

富山県立大学　電子・情報工学科
1515050　山本 聖也
指導教員　奥原 浩之 教授
平成30年2月6日

目 次

図一覧

表一覧

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
クラスター	X，Y
クラスター内での重心とサンプルとの距離の 2 乗和	$\mathrm{L}(X)$，$\mathrm{L}(Y)$
クラスターの重心とクラスター内の各サンプルとの距離の 2 乗和	$\mathrm{L}(X \cup Y)$
入力データベクトル	x
出力層のニューロンの番号	i
参照ベクトル	m_i
勝者ニューロン	c
勝者ニューロンとの距離によりガウス関数で減衰する係数	h_{ci}
i 番目のニューロンの出力層上での位置	r_i
勝者ニューロンの出力層上での位置	r_c
学習回数	t
学習率係数	$\alpha(t)$
学習半径	$\sigma^2(t)$
位置	x
速度	v
運動量	w
0 から 1 の乱数	r
調整パラメータ	c
各個体の過去の最良個体	x_{db}
集団中の最良個体	x_{gb}
ステップ幅	ϕ
粒子	P
固有値	λ
PSO の調整パラメータ	$\alpha, \beta, \gamma, \delta$
ニューラルネットワークのダイナミクスに由来する新しい行列	X_μ
勾配情報	∇E

第1章

序論

§ 1.1　本研究の背景

近年，スマートフォンを始めとする様々なデバイスを身に着けることが一般的であり，情報技術の急速な発展が著しい．中にはウェアラブルデバイスと呼ばれるものがある．それらは個人の細やかな行動をデータとして蓄積し，記録しアプリケーションを介することにより快適なサービスをユーザーに提供するとともに，健康面の管理なども行ってくれるものも存在する．このようにスマートフォンやウェアラブルデバイスを使用して取得した行動のデータは，個人の生活に活かしたり，社会に活かしたりできると考えられている．

スマートフォンやウェアラブルデバイスの全地球測位システム (Global Positioning System, Global Positioning Satellite : GPS) から個人の位置情報を取得，解析するアプリケーションが多く存在し，スケジュール情報と合わせることによりコミュニケーションツールとして活用されたり [1]，受容性の高いライフログの研究が行われている [2]．また，気温や湿度などのユーザーの周囲の環境情報と，ユーザー自身の体温や心拍数などの生体情報をログとして蓄積することにより，運動時の快適さ [3]，運転状態推定技術の開発 [4]，人体活動のモニタリングシステム [5] など，ユーザーの役に立つ情報の提示や，サービスの開発を行うことができる．

今の日本は高齢化問題，災害大国といった問題を抱えている．高齢化が進めば常時健康管理が必要な高齢者も多くなるだろう．そこでウェアラブルデバイスから環境・生体情報を管理することにより，高齢者が誤って引き起こしてしまう事故の事前の対策，看護師を始めとした医療従事者の業務の軽減が可能となる．災害時には位置情報を活用した避難システムを開発することにより，災害に対して事前に対策を講じることができる．

しかし環境ログや生体ログといったものは，データによっては精密な個人情報が含まれるため，不安や嫌悪感を感じる場合もあり，情報漏えいのリスクへの警戒など，技術面とは異なった問題も存在している．[6]．また，手動でライフログデータを取得するアプリケーションも多く，未だライフログの受容性は改善の余地がある．

環境・生体ログデータを取得する際にユーザーの負担を軽減するためにリアルタイムかつ迅速なウェアラブルデバイスの開発が必要であり，また従来法よりも有効な解析手法の

提案を行う必要があると考えられる．開発するデバイスは一般的なデバイスよりも多くのデータを取得できるべきであり，膨大なデータであっても高い精度の行動の識別が行えることが理想である．

§ 1.2　本研究の目的

本研究は，従来よりも多くの情報収集を可能とした環境・生体ログシステムの開発，その情報をもとにした行動パターンの類似性やイベント性を検出・考察することを目的とする．一般的に使用されているウェアラブルデバイスやアプリケーションについて述べ，問題点を考察したうえでシステムの開発を行う．開発するシステムは無線通信から自動でのデータの蓄積を行うことにより，ユーザーへかかる負担を少なくしている．また，システムの開発にあたって，マイコンと多くのセンサを使用している．取得したデータはリアルタイムでの管理を行えるように，ブラウザ上で折れ線グラフのリアルタイムプロットを行えるシステムも加えて開発する．

行動パターンの類似性やイベント性の検出には従来手法によるクラスタリングを行う．従来手法によるクラスタリングではテキストデータが用いられているが本研究で扱うデータは数値データであるため，テキストではなく数値のクラスタリング結果を示す．示す結果は，自己組織化マップ（Self Organizing Maps : SOM），階層的クラスター分析，多次元尺度構成法（Multi Dimensional Scaling： MDS），対応分析，共起ネットワークを行い，読み取りを行うことで環境・生体ログデータの類似性やイベント性を考察擦る．

従来よりも優れた解析手法の開発を目的として，粒子群最適化 (Particle Swarm Optimization : PSO) に勾配・制約を考慮したハイブリッド的な応用手法を定式化するとともに有効性を示す．またその手法を用いた PSO のクラスタリング手法を提案する．

また PSO クラスタリングの比較対象として，自己組織化マップ（Self Organizing Maps : SOM），階層的クラスター分析，多次元尺度構成法（Multi Dimensional Scaling： MDS），対応分析，共起ネットワークを行い，読み取りを行うことでライフログデータの類似性やイベント性を考察したうえで，提案手法のアルゴリズムを示す．

§ 1.3　本論文の概要

本論文は次のように構成される．

第 1 章：本章 第 1 章では，本研究の概要と目的について説明した．

第 2 章 第 2 章では，現状のライフログ・ライフログアプリケーションの問題や特徴について説明する．また，データの取得に用いるデバイスやセンサについて説明する．

第 3 章 第 3 章では，行動識別についての研究や，行動識別のための分析手法について説明する．また，分析から考えられる類似性やイベント性について説明する．

第 4 章 第 4 章では，一般的な PSO に勾配情報を組み込んだハイブリッドな PSO についての定式化について述べる．また PSO を用いたクラスタリングについて解説し，提案手法のアルゴリズムを説明する．

第 5 章 第 5 章では，まとめと今後の課題を述べる．

第2章

ライフログと各種センサ

§ 2.1　現状のライフログ

ライフログ（lifelog) とは，人間の活動（life) の記録（log) であり，センサーなどで個人の活動に関するログを取得する行為が，ライフログの語源と考えられている [7]．本研究では，この行為をライフログとし，個人の行動履歴に基づいて生み出されるビッグデータのことをライフログデータと呼ぶこととする．また，ライフログに関して，長時間の記録や膨大なデータが必要という定義はない．

ライフログデータを取得・活用できるアプリケーションとして，iphone の専用アプリ「ヘルスケア[1]」がある (図 2.1 参照)．このアプリケーションはアクティビティ[2](図 2.2 参照) から，どれほど歩いたかという歩数や消費エネルギー等のライフログデータを取得し，ユーザ自身が健康管理に生かすことができる．また，自動で位置情報をマップにマッピングできる「マッピング - GPS ログまとめて全部記録[3]」(図 2.3 参照) や，手動でマッピングする「Swarm[4]」(図 2.4 参照) というアプリケーションがある．このアプリケーションは行動の記録を取ることができるため，日々の生活や旅行の記録として使用できる．上記のアプリケーションは GPS のアクセス許可が必要であり，上記以外のライフログアプリケーションも GPS を必要とすることが多い．

ライフログに関する既存研究として，スマートフォンから得られる位置情報履歴や写真撮影履歴，ツイートを使用したライフログデータから行動特徴抽出・イベント検出を行う研究が行われている [8] [9]．取得したライフログデータの解析を行うことで，ユーザー自身の健康管理や学習 [10] に生かすだけではなく，ビジネスとしてターゲティング広告に生かすこともできる．ライフログは，様々な視点からライフログデータの比較を行うことで，個人や社会に利用できるという価値があると考えられる．

しかし，現状のライフログには，大きくわけて二つの問題があると考えられる．一つ目

[1]https://www.apple.com/jp/ios/health/
[2]https://appllio.com/how-to-use-iphone-healthcare
[3]https://play.google.com/store/apps/details?id=org.liteapp.mat2
[4]https://play.google.com/store/apps/details?id=com.foursquare.robin

図 2.1: ヘルスケア[1]

図 2.2: アクティビティ 2[2]

はライフログの多様化問題，二つ目はライフログの煩雑問題である．

ライフログの多様化問題

ライフログを扱うサービスとしてインターネット上には多種多様なアプリケーションが登場し，様々な種類のライフログを Web 上で確認できる [?]．ユーザーの考えや気持ちを記録するブログや Twitter[5]，写真を記録する Flickr[6]，三度の食事を記録する FoodLog[7]，体や運動のデータを記録するからだログ[8]，携帯電話で写真やバーコード，睡眠時間を記録するねむログ[9]などが存在する．このように一つにライフログといっても様々な形で表すことができ，それぞれによって用いるデータは全く違うものになっている．　上記のようにライフログが多種多様になっている一方で，多くのアプリケーションを並行して使用させることはユーザーへの負担となることも考えられる．またアプリケーションが増えることによりユーザーが扱う情報量も増加し，すべての情報を正確に管理することが厳しくなってしまう．

ライフログの煩雑問題

ライフログアプリケーションの中には，意識的にライフログデータを取得しなければならないアプリケーションが存在する．このようなアプリケーションはライフログのために，ユーザーが自ら位置情報をマッピングしたり，食事風景の写真をとることを意識しなくてはならない [12]．ユーザーの主観的なライフログデータを取得できるが，ライフログデータを取得するのに手間がかかってしまうという問題を引き起こす．

[5]https://twitter.com/
[6]https://www.flickr.com/
[7]https://www.foodlog.jp/

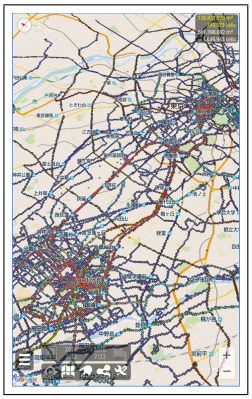


図 2.3: マッピング-GPS ログまとめて全部記録[3]

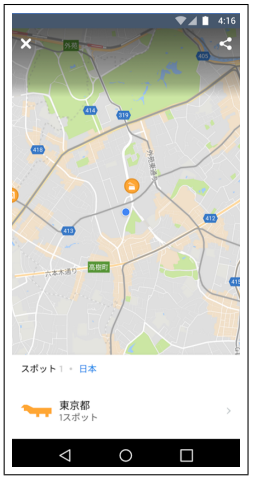


図 2.4: Swarm[4]

また，ライフログの個人情報問題に関して，株式会社ＮＴＴデータ経営研究所が 2016 年に 10 代から 60 代の男女 1059 人を対象として実施した「パーソナルデータに関する一般消費者の意識調査 [13]」という調査がある．この調査において,「企業のマーケティング等の利用目的にて，パーソナルデータを企業に提供しても良いと思うデータの条件」において，金銭や商品を受け取ることができたり，個人が特定できない状態でも，どのような条件であっても位置情報は提供したくないという人が 66.2%であり，過半数以上を占めていることがわかっている．

ライフログの多様化問題，煩雑問題という二つの問題に対し，ライフログデータを収集する上で重要であることは，一つのアプリケーションで多くのデータを取得し，ユーザーにサービスとして提示するまでの処理を手間をかけずに無意識に行うことであると考える．また示す結果に誤った情報をはじめとしたノイズが含まれてはいけない．よって画期的な手法によるライフログを正確にクラスタリングすることが重要であると考えている．

§ 2.2　各種センサとマイコンの概要

近年，スマートフォンやタブレットなどのスマートデバイスが急速に普及している．その次のデバイスとして期待されているものがウェアラブルデバイスである．ウェアラブルデバイスとは，体に装着して利用するコンピュータデバイスの総称であり，代表的なものとして，一般で使われている Apple の「Apple Watch[10]」(図 2.5 参照) や，医療や介護の現場で用いられている EMC Healthcare の「CALM-M[11]」(図 2.6 参照) などが挙げられる．

このようなウェアラブルデバイスは日常的な場面から，医療や介護といった幅広い場面で用いられている．また, ウェアラブルデバイスから得られる使用者の生体情報や個人情報は行動認識技術に活かされている．よって行動認識技術の応用範囲はとても広く，軍事 (兵

[8]https://help.goo.ne.jp/goo/g108/
[9]https://nemulog.co.jp/
[10]https://www.apple.com/jp/watch/
[11]https://prtimes.jp/main/html/rd/p/000000003.000024862.html

図 2.5: Apple Watch[10]

図 2.6: CALM-M[11]

隊，整備士)，業務 (営業マン，消防，警察，飲食店，コンビニ，警備，介護)，民生 (情報開示，記憶補助，コミュニケーション，エンタテイメント，教育) などの場面での利用を考えることができる [?].

さらに，今ではほとんどの人が持ち歩いていると言えるスマートフォンにも多くのセンサが搭載[12]されており，スマートフォンのアプリケーションを介することにより，特に必要な操作や手間なく，生体情報をはじめとした情報を得てライフログとして蓄積し，それらをユーザー自身で管理・確認することができる．

本研究では，Arduino, LLC 社の「Arduino UNO」((図 2.7 参照) や，Raspberry Pi Foundation による「Raspberrypi3.0」(図 2.8 参照）などのマイコンと，脈拍や体温などを測る生体センサや，気温，湿度などを測る環境センサ，これらをウェアラブルデバイスとして使用する．使用する理由としては，Arduino や Raspberrypi を用いることにより扱うことのできるセンサが幅広く，センサの追加や撤去などの調整が容易に行えるためである．

生体情報として用いるセンサは，体温[13]，心拍[14]，GSR(Galvanic Skin Response)[15]である．体温や心拍は医療や介護の現場で用いられており，ユーザーの体調管理に欠かせない．GSR は客観的に人間の心理を評価をするなどできるので感情を分析できる特徴を持つ．

また加速度[16]，温湿度気圧[17]，照度[18]，GPS[19]，人感[20]といった環境情報センサを用いる．これらのセンサと生体センサを組み合わせ，周囲の情報を取り入れることにより，ユーザーがどんな状況でどんな行動をしているか，どこでどんな行動をしているかというように，より細かい情報を得ることができる．また加速度を活かすことにより，姿勢推定や，歩行・転倒の検出など行動検知が可能となる [?] [?]．また位置情報は Google 社による「Google maps[21]」(図 2.9 参照) を代表として歩行，運転時の目的へのナビゲートと日常に利便性をもたらしたり，災害などの緊急時の行方不明者や徘徊老人の捜索に用いられることもある．人感センサは赤外線照度センサとも言われており，周囲の人肌以上のものを感知すること

[12]https://garumax.com/smartphone-sensor
[13]http://naritaku.hatenablog.com/entry/2016/04/05/230649
[14]http://myct.jp
[15]http://wiki.seeedstudio.com/Grove-GSR_Sensor/
[16]http://akizukidenshi.com/catalog/g/gK-13010/

図 2.7: Arduino UNO

図 2.8: Raspberrypi 3.0

図 2.9: Google maps[21]

ができるので，ユーザーが他人と接しているかどうかを判別することができる [?].

ここまで挙げたセンサをまとめると以下のようになる．

取得するデータ一覧

・生体センサ
体温，心拍，GSR
・環境センサ
GPS(緯度，経度，海抜)，温度，湿度，気圧，照度，人感，加速度 (3 軸)，角速度 (3 軸)，磁気コンパス (3 軸)

上記のセンサを Arduino を用いて計測する．Arduino に計 8 つのセンサを接続し，20 種類のデータを計測できる環境を作成する (Arduino とセンサの画像).

§ 2.3　無線によるセンサデータ収集

ライフログを取得をするにあたって，2.2 節で挙げたマイコン (Arduino，Raspberrypi) とセンサ類を組み合わせて独自のライフログの測定環境を開発する．ライフログを取得する

[17]https://github.com/SWITCHSCIENCE/BME280/blob/master/Arduino/BME280_I2C/BME280_I2C.ino
[18]jkoba.net/prototyping/arduino/cds_practice.html
[19]https://www.petitmonte.com/robot/howto_gysfdmaxb.html
[20]http://tech.blog.surbiton.jp/arduino_motion_sensor_se-10/
[21]https://www.google.co.jp/maps/?hl=ja

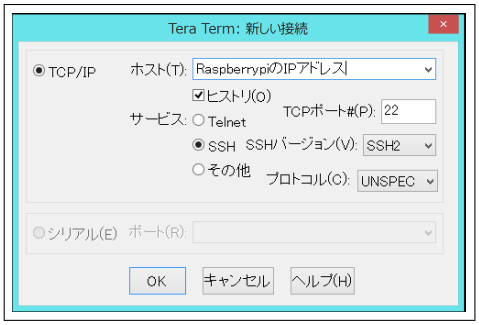


図 2.10: Tera term[22]

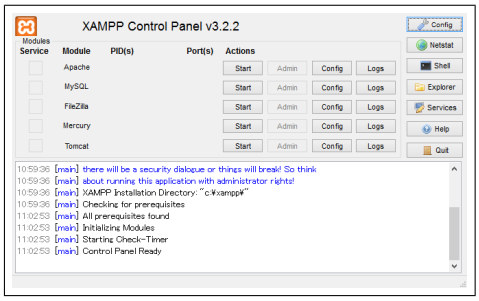


図 2.11: XAMPP[23]

にはユーザーに手間がかからずスムーズかつ無線通信を利用した IoT の側面も考慮した環境を作成する必要がある．

手法としては Arduino に必要なセンサを繋ぎ，さらに Arduino と Raspberrypi を USB ケーブルで有線接続し，Arduino と Raspberrypi でシリアル通信を行う．これにより Arduino で取得したライフログが Raspberrypi に送信される．Raspberrypi には Arduino と違い Wifi 環境が整っているので無線通信を行うことができるので，Arduino から送られてきたライフログを PC に無線通信で送り，PC 内でそのライフログを蓄積する．

Raspberrypi と PC の接続は寺西高氏によって開発された Tera term[22] という Windows 上で動作する多機能端末を用いる．Windouw から Linux などにリモート接続をしようとするときによく用いられている．Tera term を起動すると IP アドレスを指定して任意の端末とリモート接続できる (図 2.10 参照)．リモート接続を終え，Raspberrypi 上で Python のプログラムを作成し，Arduino から送信されたライフログを PC に送信する．

Raspberrypi からライフログを送信するにあたって，XAMPP[23] という完全無償の MariaDB，PHP および Perl を含んだ，Apache ディストリビューションを利用する (図 2.11 参照)．これを用いてサーバーとして PC 上で Apache でローカルサーバーを作成する．これで PHP の開発環境を容易に使用することができる．

しかし，この手法では Arduino と Raspberrypi の二つのマイコンを使用しなければならないので，日常の情報を集めることが目的のライフログではユーザーに対して負担がかかってしまう．なのでライフログを測定する端末はよりコンパクトでスペースを取らないものの方が良いと考えられる．

そこで無線通信の役割を担っている Raspberrypi に代わって Arduino 上で使用することのできる Wifi モジュール ESP-WROOM-02[24] を使用する．この Wifi モジュールを用いることにより Arduino に欠けていた無線通信機能を補うことができるとともに，Raspberrypi を使用せずともよくなるので，マイコン一個分スペースを削減することができ，小型・軽量化することができる．

Wifi モジュールを用いる場合においてのライフログの処理は Raspberrypi の場合と同様に，PC 上で作成したローカルサーバーにライフログを送信する．

また 2.2 よりセンサの数を合計すると 20 次元のデータとなる．

[22]https://ja.osdn.net/projects/ttssh2/
[23]https://www.apachefriends.org/jp/index.html
[24]https://www.amazon.co.jp/exec/obidos/ASIN/B01C8ANPYW/vlsiprograma-22/

環境・生体ログはPC上に蓄積することができるが，データとして蓄積するだけでは意味が無い．ユーザーの情報を取得し，データをリアルタイムでサービスとして反映させることが，実際のアプリケーションでは必要である．今回はユーザーが自身の情報を視認できるように環境・生体ログの簡易的な可視化を行う．

ローカルサーバー状のデータをリアルタイムで送信，表示させるためにSocket.IO[25]による通信手法を用いる．Socket.IO[25]とはweb上においてリアルタイムの通信を実現する技術の一つである．これを用いて送信されてきたデータをhtmlで作成したページに送信する．ウェブサイトに送るためにnode.js[26]をインストールしたPCで以下のプログラムを実行しサーバーを立てる．

```
//サーバー側
var http      = require('http');
var socketio = require('socket.io');
var fs        = require('fs');
var server    = http.createServer(function(req, res) {
    res.writeHead(200, {'Content-Type' : 'text/html'});
    res.end(fs.readFileSync(__dirname + '/index.html', 'utf-8'));
}).listen(3000);

var io = socketio.listen(server);

io.sockets.on('connection', function(socket) {
  console.log('connection');
    socket.on('bio_csv',function(msg){socket.broadcast.emit('q',msg);});//ラ
    socket.on('bio2_csv',function(msg){socket.broadcast.emit('w',msg);});//
});
```

このサーバーを経由してページに送信し，その結果は適当なブラウザ上で確認できる．以下にその結果の例を示す(図2.12参照)．この例では送信しているデータは体温と心拍である．

データの取得からブラウザ上での可視化までのデータフローを図にまとめる（図2.13参照)．ArduinoからXAMPP[23]のApacheで立てたローカルサーバーを用いてPHPファイルへ，そしてaccept.jsに送信している．accept.jsでシステムを2秒程度停止させることによりプログラムが正しく動作する．2秒の停止がない場合では正しく動作しない．その後PC上で立てたサーバーからindex.htmlというファイルに送信し，ブラウザ上でデータが確認できる．データの蓄積はPHPファイル上でcsv形式で保存を行っている．

htmlでデータをグラフ化しつつリアルタイムプロットを行うにはEpoch.js[27]を用いた．選んだメリットとしては，リアルタイム表示に特化していて設定が簡素であるからである．以上の手法によりArduinoとセンサで取得したデータをライフログとして蓄積，表示できるようになった．

[25]https://ics.media/entry/4320
[26]https://nodejs.org/ja/
[27]https://qiita.com/okoppe8/items/d8d8bc4e68b1da4a0a36

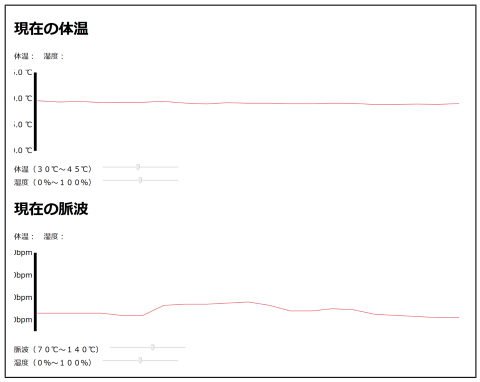


図 2.12: データの可視化[25]

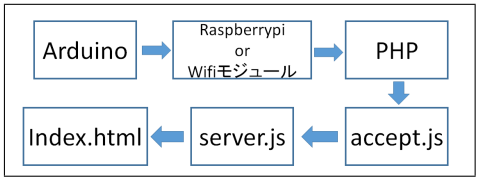


図 2.13: データフロー[26]

第3章

センサデータからの行動識別

§ 3.1　行動識別

携帯電話やウェアラブルデバイスを用いて，ユーザーが今何を行っているかという行動をライフログデータとして取得し，取得したライフログデータの解析から行動を認識することを行動認識や行動識別という．本研究では，行動識別と呼ぶことにする．

既存研究には，携帯電話の加速度センサや GPS を用いてライフログデータを取得し，走行や歩行しているなどの行動識別を行う研究 [18] や，ウェアラブルデバイスの加速度センサや GPS を利用する事で人の行うさまざまな行動を取得し，行動識別を行う研究 [19] がある．しかし，本を読んでいることであったり，料理をしていることなどの細かい動作をライフログデータとして取得することは難しい．細かい動作をライフログに組み合わせるため，手動で動作の開始・終了を記録するアプリケーション「行動の記録 (LifeLog)[22](図 3.1 参照)」や，机上に設置した Kinect™(図 3.2 参照) を用いて机上の細かい動作を認識する研究がある．しかし，この研究は机上に限っているため屋外や机上以外の行動は認識できない [20]．なお，Kinect™ は 2017 年 10 月 25 日に生産終了が公表されている．

本研究では 2 章で述べたように Arduino とセンサ類を使用して，細かい行動をライフログデータとして取得する．ユーザーの生体情報，周囲の環境情報，位置情報を取得することにより，ユーザーの体調管理やどのような行動をしているか，周囲が厚いのか寒いのか，どこにいるかなどを把握することができる．生体データと加速度データを用いての行動識別は [?] で行われているが，本研究では 20 次元によるデータを扱うのでこういった従来法より，より多種類の行動識別を行うことができると考えられる．

§ 3.2　行動識別のための分析手法

本研究ではいくつかの解析手法を用いて，一定時間内の取得データを視覚的に表し，行動識別を行う．そのために，多変量解析である SOM，階層的クラスター分析，MDS，対応

[22]https://play.google.com/store/apps/details?id=com.yoko.tama.workLog&hl=ja

図 3.1: 行動の記録 (LifeLog)[10]

図 3.2: Kinect™

分析，共起ネットワークを用いてテキストデータの可視化を行う．

多変量解析を行うツールとして KH Coder[23]を使用する．KH Coder とは，テキスト型データの計量的な内容分析，もしくはテキストマイニングのためのフリーソフトウェアである [22]．無償でウェブサイトから入手でき，すべての機能をマウス操作で利用できる．また，どんな言葉が多く出現していたのかを頻度表から見ることができたり，SOM，共起ネットワークなどの多変量解析を行ったりできる [23]．KH Coder を用いて行われた研究としては，アンケートの自由回答項目・新聞記事・インタビューデータなどさまざまなデータを分析した事例がある [24]．本研究では Version 3.Alpha.11 を使用する．

データは 2 節で述べたセンサから取得する数値データを扱う．KH Coder はテキスト型データに対しての分析，もしくはテキストマイニングを目的としているため 2 節で述べたようなセンサの数値データを入力としても数値データが読み取られない．なので Arduino から取得したデータを分析を掛ける前に記録させておく必要がある．KH　Coder の語の取捨選択より強制抽出する語の指定を行う．図の語は全体の一部で緯度データを指定している部分である (図).

KH Coder をダウンロードする際に取得できる KH Coder3 リファレンス・マニュアルによると，KH Coder を使用した多変量解析には，まず対象のテキストファイル（もしくはエクセルファイル）を読み込む事から始める．今回はセンサデータを取得する際に作成した CSV ファイルから読み込む．読み込む CSV ファイルに事前に手動で h1 タグや h2 タグという見出しタグを設定することで，見出しごとの解析も可能である．今回はデータを取得する際にデータの先頭にそれぞれ文字をつけてそれぞれの数値との見分けがしやすくなるようにした (緯度なら LON，経度なら LAT)．そのデータの一覧を示す (表) 次に，前処理として，POS Tagger[25]を使用して自然言語処理を行う（表 3.1 参照）．前処理を行うことで，多変量解析に使用する「各文書に，それぞれの語が何度出現していたのかという集計表」である「文書×抽出語」表 [25](表 3.2 参照) を出力することができる．h1 から h5 とい

[23]http://khc.sourceforge.net/

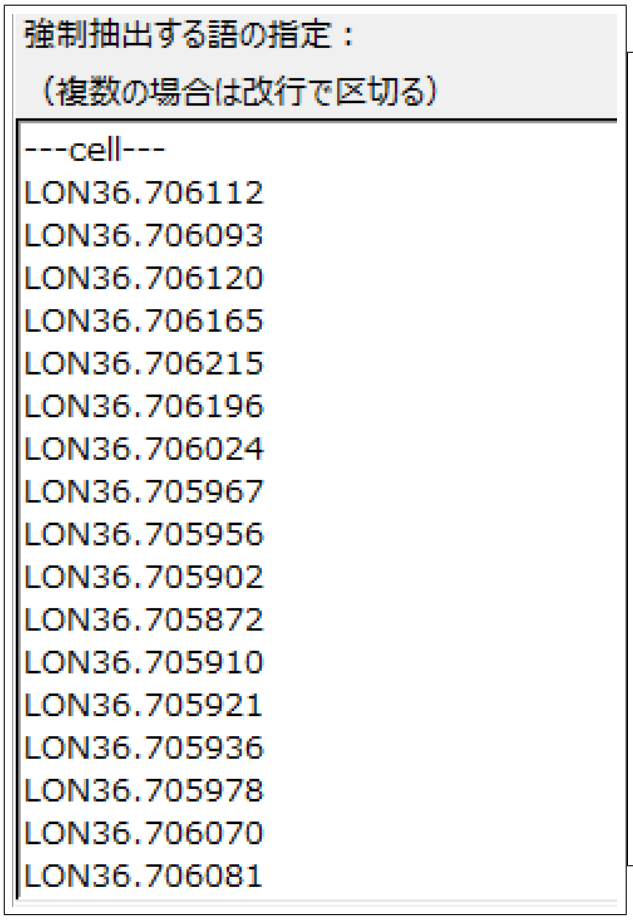


図 3.3: 強制抽出する語の指定の一部

図 3.4: Stop words の一部[24]

うのは見出し番号であり，h1 タグや h2 タグが存在すると 1 増加する．dan は段落番号で，bun は文番号である．id は文書の通し番号で，リセットされることは無い．length_c は文書の長さを文字数で表し，length_w は文書の長さを語数で表したものである．

まず，ライフログデータの内容解析のため，階層的クラスター分析，MDS，対応分析，共起ネットワークを行う．この時テキストデータは，抽出語の中でも，多く出現する 29 語の抽出語を用いる．

階層的クラスター分析は，抽出語の最も似ている組み合わせから順番にクラスターにしていく方法であり，デンドログラムを表示する [26]．指定されたクラスター数に全体を分割し，その結果を色分けによって表示する．なお，KH Coder では，デフォルトの Auto では，抽出語数の平方根を四捨五入したものを用いている [27]．1 つのクラスターには関連性が高い抽出語が集まっているため，クラスターごとに集まっている抽出語を調べることでテキストデータ全体における文書の傾向や特徴を知ることができる．

階層的クラスター分析の作成方法は，まず抽出語として A, B, C, D があったとする．この時抽出語の中で最も距離の近い組み合わせを A と B とし，A と B をくくり，2 点の代表点を求める．次に，AB の重心，C，D の 3 点で，最も距離の近い組み合わせを見つける．このとき C と D が最も近いとすると，C と D をくくる．このように繰り返していくことで，

[25]https://nlp.stanford.edu/software/tagger.html

表 3.1: 取得したセンサの数値データの一部その 1

緯度	経度	海抜	気温	湿度	気圧	照度	人感	加速度 x
LON36.706112	LAT137.096690	HEI29.8	T28.77	HUM31.97	1	26	5	0
LON36.706093	LAT137.096620	HEI29.7	T28.84	HUM31.54	2	684	163	0
LON36.706120	LAT137.096540	HEI29.4	T28.88	HUM31.67	3	876	213	0
LON36.706165	LAT137.096540	HEI28.8	T28.91	HUM33.01	4	666	163	0
LON36.706215	LAT137.096540	HEI27.7	T28.84	HUM32.97	5	344	88	0
LON36.706215	LAT137.096620	HEI26.2	T28.90	HUM31.81	6	469	119	0
LON36.706215	LAT137.096630	HEI24.2	T28.91	HUM32.35	7	470	131	0
LON36.706196	LAT137.096630	HEI21.7	T28.92	HUM32.30	8	587	148	0
LON36.706196	LAT137.096620	HEI18.9	T28.89	HUM33.02	9	409	115	0

表 3.2: 取得したセンサの数値データの一部その 2

加速度 z	角速度 x	角速度 y	角速度 z	磁気 x	磁気 y	磁気 z	体温	心拍	GSR
0	0	0	0	1	1	26	5	0	0
0	0	0	0	2	2	684	163	0	0
0	0	0	0	3	3	876	213	0	0
0	0	0	0	4	4	666	163	0	0
0	0	0	0	5	5	344	88	0	0
0	0	0	0	6	6	469	119	0	0
0	0	0	0	7	7	470	131	0	0
0	0	0	0	8	8	587	148	0	0
0	0	0	0	9	9	409	115	0	0

デンドログラムを作成する [26] [28].

KH Coder3 リファレンス・マニュアルによると，クラスター間の距離測定方法として，KH Coder ではウォード法を使用している．2 つのクラスター X,Y を結合したと仮定したとき，それにより移動したクラスターの重心とクラスター内の各サンプルとの距離の 2 乗和 L(X ∪ Y) と，もともとの 2 つのクラスター内での重心とそれぞれのサンプルとの距離の 2 乗和 L(X)，L(Y) の差が最小となるようなクラスターどうしを結合する手法である．ウォード法は，計算量は多いが分類感度がいいため用いられることが多く，ウォード法は一つのクラスターに抽出語が順に吸収され類似するクラスターが形成される鏡効果が起こりにくいという強みがある [29] [30].

$$\Delta = L(X \cup Y) - L(X) - L(Y) \tag{3.1}$$

クラスター分析の結果を図 3.5 に示す．この時クラスター数を Auto にしたためクラスター数は 5 となる．併合標準（図 3.6 参照）からクラスター数が 5 であることは妥当だと考えられるためクラスター数は 5 とした．クラスター分析から，最も多く出現している ANNE という単語やほかにも多く人名と思われる単語があることからある程度主要人物が予想できる．また，あるクラスターに school という単語と Avonlea という単語がある点から，school

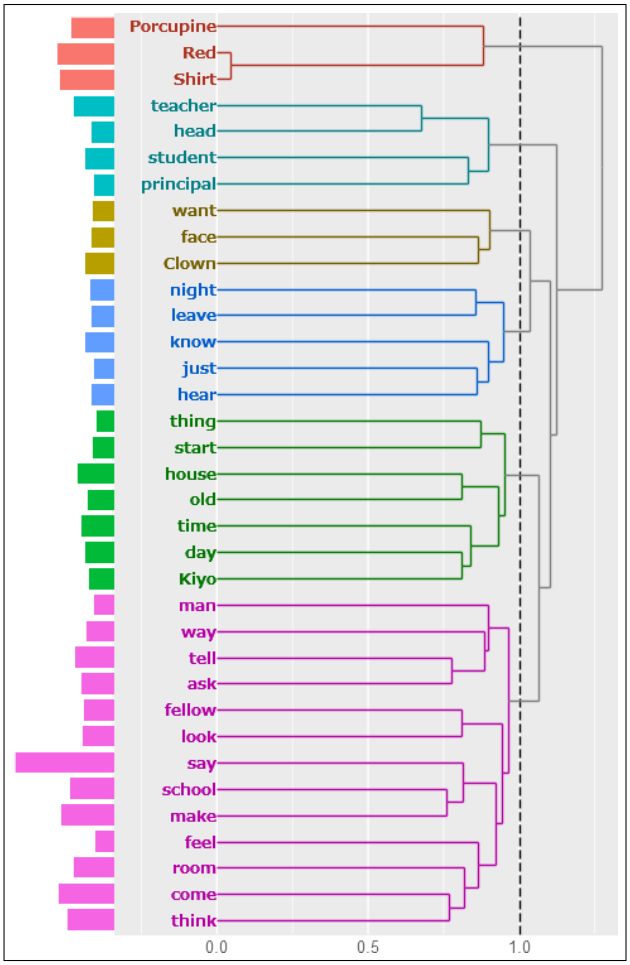


図 3.5: センサデータから作成したクラスター分析

は Avonlea という地名のところにある，または深く関係していることが分かる．クラスター内やクラスター同士の比較からデータ内で重要な語の関係を推測できる．

MDS は，抽出語間の関連性や類似性の強さをマップ上の点と点の距離に置き換えて，相対的な関係性を視覚化する手法である [31]．KH Coder では MDS の中でも最も広く利用されてきた Kruskal の非計量 MDS を使用している [32]．また，語と語の関連を見るために Jaccard 係数を使用している．Jaccard 係数とは二文章間の類似度であり,「語 A を含む」かつ「語 B を含む」文書の数を,「語 A を含む」または「語 B を含む」どちらかでも当てはまる文書の数で割った係数である [33]．MDS の結果は，相対的な位置関係だけを表わしているため軸の方向性に意味はない．

$$Jaccard\,係数 = \frac{語\,A\,と語\,B\,を含む文書}{語\,A\,もしくは語\,B\,を含む文書} \tag{3.2}$$

図 3.7 は赤毛のアンデータから出力した MDS である．クラスター分析を参考にし，クラスター数は 5 に設定した．この結果，目立つものはクラスター 03,04 であり，01,02,05 と比べて他のクラスターとそれほど離れてないことから，データの中で中心的なクラスター・抽出語であることがわかる．クラスター 02 と 05 のように離れて配置されるクラスターもあ

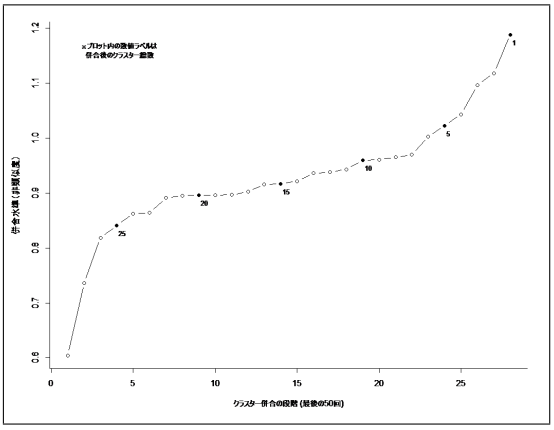


図 3.6: クラスター分析の併合標準

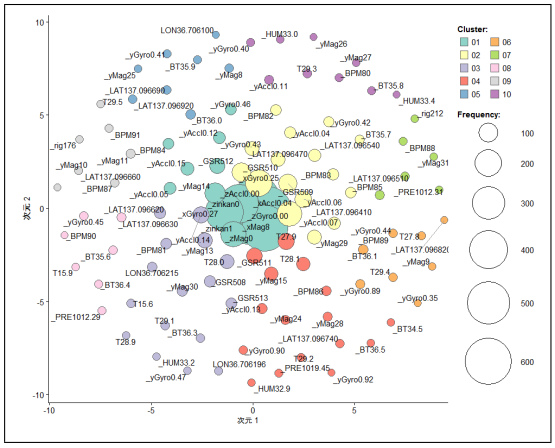


図 3.7: センサデータから作成した MDS

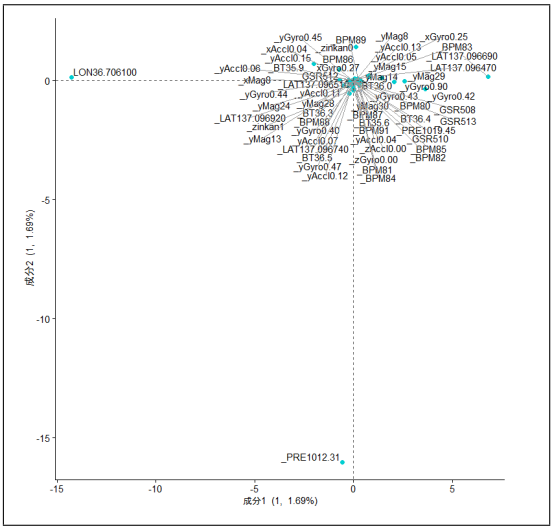


図 3.8: センサデータから作成した対応分析

り，関係性の低いクラスター・抽出語であることがわかる．

対応分析は，単純な 2 次元表や多重表の行と列間の対応する測定値を分析する探索的データ解析の手法であり，分析結果として，2 次元のマップが表示される [34]．このマップで近くに位置しているものは，相対的に関連が強いということを示し，遠くに位置しているものは関連が弱いということになる．また，対応分析では，これといって特徴のない語が原点付近に密集することが多い．この時軸に表示されている数字は固有値と寄与率である．

図 3.8 は赤毛のアンデータより出力した対応分析である．この対応分析から，head や girl や MARILLA という単語は特徴的ではなく，データ全体によく出現することがわかる．一方で，Jane や evening は原点から遠く離れているため，特徴的であることがわかる．

§ 3.3　類似性・イベント性

本研究では，ライフログデータの内容解析のため，階層的クラスター分析，MDS，対応分析，共起ネットワークを行った後，データの時系列をSOMを用いて解析を行う．SOMを用いて，テキストデータの時系列を可視化することでテキストデータの類似性を検出する．

SOMとは，ヘルシンキ大学のコホーネン教授により1981年頃に発表された，教師なし学習を行なうニューラルネットワークの代表例と言える解析手法である [39]．ニューラルネットワークとは，脳機能に見られるいくつかの特性を計算機上のシミュレーションによって表現することを目指した数学モデルである．つまり，人間が無意識にやっていることを機械にやらせるということである．

SOMは図??に示すように入力層と出力層の2つに分かれて競合学習を行う [40] [41]．入力層のニューロンが複数個あるが，各々のニューロンがそれぞれの次元に対応した出力を行っていると考える．入力データベクトルと呼ばれる入力層から出力層への入力を x と定義し，出力層のニューロンと入力層のそれぞれのニューロンとの結合強度は総称して参照ベクトルと呼ばれ，i を出力層のニューロンの番号とすると，m_i で表される．まず初めに，m_i の初期化を行い，入力データベクトル x を選び，入力データベクトルと各ニューロンの参照ベクトルとのユークリッド距離で出力層のニューロンを競合させる．勝者ニューロンを c とすると，式3.3で表される．$\arg\ \min\ f(a)$ は $\mathrm{f}(a)$ を最小にする a の集合であり，下側に変数がとる値の範囲を書くことが多い．

$$c = \arg\min_i \{||x - m_i||\} \tag{3.3}$$

次に，勝者ニューロンと勝者ニューロンに近いニューロンは自らの参照ベクトルと入力データベクトルを近づける学習を行うため，参照ベクトルを同様に更新させる．この時，h_{ci} は勝者ニューロンとの距離によりガウス関数で減衰する係数である．

$$m_i(t+1) = m_i(t) + h_{ci}(t) \cdot \{x(t) - m_i(t)\} \tag{3.4}$$

$$h_{ci} = \alpha(t) \cdot \exp \frac{-||r_c - r_i||^2}{2\sigma^2(t)} \tag{3.5}$$

また，SOM作成過程ではユークリッド距離を利用している．また，KH Coder3 リファレンス・マニュアルによると，文書の長さのばらつきに左右されない形で計算を行うために，文書中における語の出現回数をそのまま使うのではなく，1,000語あたりの出現回数に調整したものを計算に使用している．

KH Coder3 リファレンス・マニュアルによると，KH CoderのSOMの学習は，大まかな順序づけを行う段階と，微調整を行う収束段階の2段階で行われる．KH Coderでは，1段階目が1,000，2段階目が「全体のノード数を500倍した数値」に設定されており，全体のノード数が40の場合は200,000回である．また，各ノードがもつベクトルをウォード法で分類してクラスター化する，クラスター数は任意に決定できるため，クラスター分析などの結果からクラスター数を調整する．

本研究ではこのKH Coderはデータの前処理段階に使用し，実際のSOM作成には，データ解析・グラフィックス環境を備えたオープンソースのソフトウェアであるRを使用する [42]．

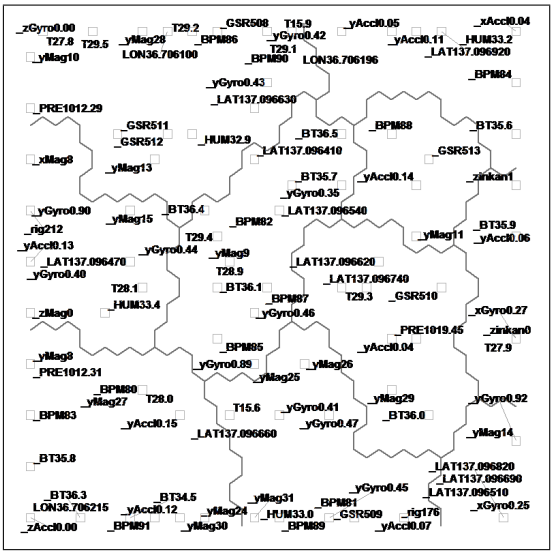


図 3.9: センサデータから KH Coder で作成した SOM

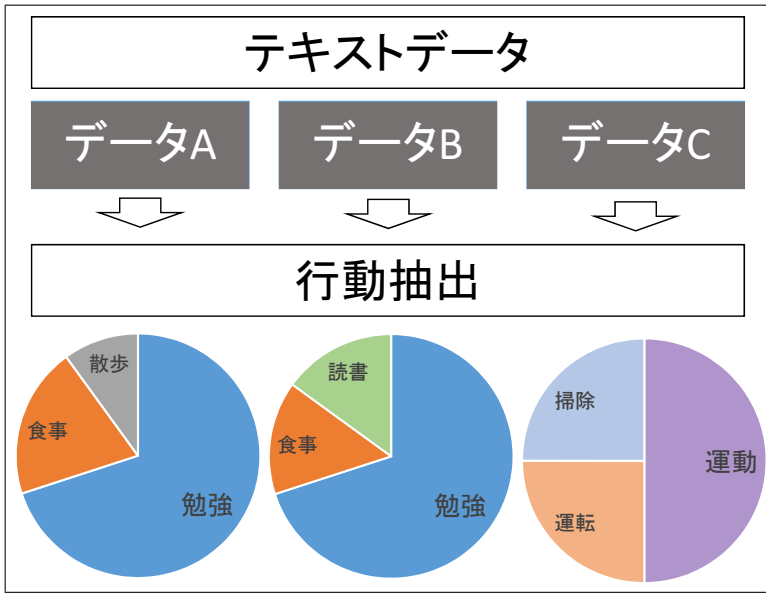


図 3.10: データ A, データ B, データ C の例

KH Coder で出力できる SOM（図 3.9 参照）は抽出語どうしの関係を示すものとなっているため，今回のライフログデータの解析に用いることには向かないためである．

KH Coder で出力できる SOM の R ファイルを基にソースコードを書き換え，R で出力を行う．出力した SOM が図??になる．この時 SOM 上の数字は id であり，文書同士の関係が表されている．学習回数は 1 段階目が 1,000，2 段階目が 200,000 となっている．また，クラスター数は 6 とした．

図??より，文書どうしにまとまりは少ないことがわかる．これは文書数が多いことと，対象としているデータが文学作品であることから似たような文章が並ぶことが少ないためであると考えられる．

KH Coder と R を用いてライフログデータであるテキストデータの多変量解析を行い，解析結果の読み取り・比較を行うことで一定時間内の行動識別を行い，ライフログデータの類似性やイベント性を検出できると考える．

本研究では，ライフログデータの類似性とは，多変量解析によるクラスターの分かれ方やクラスターを構成する抽出語から導き出せる行動や，プロットの関係性，SOM であらわされる時系列が類似している場合類似性があると考える．つまり同じ行動を行っていることや，その行動がデータ内で占める割合が似ていることが類似性のあるライフログデータだと解釈する．また，ユーザー自身の複数のライフログデータの中で類似性のあるライフログデータが多ければ，そのライフログデータは平常日を表していると考えられる．一方でライフログデータの類似性ではなく，ライフログデータから特徴的なイベント性を検出した場合，イベント日であることを検出できたり，平常日とは違うという危険を察知したりできる．

階層的クラスター分析，共起ネットワークはクラスターの分かれ方やデータを構成する単語の関係性からどのような行動があるのかがわかり，このとき予想できるクラスター数は SOM にも利用できる．MDS，対応分析はプロット点やプロット間隔から類似する行動

やイベント性のある行動がわかる．

SOM はクラスターの分かれ方や時系列を表すプロット順を追っていくことで行動パターンを識別し，多くのライフログの中でも類似したライフログか，特徴的でイベント性のあるライフログかわかる．

図 3.10 はテキストデータとして，データ A, データ B, データ C があり，このテキストデータから各データに三つの行動がある割合で存在していることが検出できた場合を表している．この時，データ A とデータ B は，データの大半が勉強を表す単語であることから勉強している時間が多いことが類似しているため類似性があるといえる．一方でデータ A，データ B と，データ C は類似する単語がなく，類似する行動がないことがわかる．

この三つのデータが一人のユーザーのライフログデータであれば，平常日とイベント日の比較に利用できる．もし，三つのデータがバラバラのユーザーである場合，ライフログの類似性があるユーザーどうしでコミュニケーションを促進することができたり．ライフログの類似性がないユーザーどうしの比較を行うことで行動の中で改善すべき行動を検出できたりする [43]．このようにライフログデータの類似性やイベント性の検出は様々な応用が可能であると考えられる．

第4章

提案手法

§ 4.1　勾配系を考慮したPSO

Swarm Intelligence（群知能）は，鳥や魚，アリのコロニーなどのグループの行動に基づく最適化手法である． この技術の一つである粒子群最適化が開発され，様々な研究に応用されている．しかし，粒子群最適化の収束は根拠がない． 本研究では，より良い最適解を求めるための群知能とニューラルネットワークダイナミックスの新しいハイブリッド動的システムを提案した．本節では主な結果として、粒子群最適化と勾配法のメカニズムを理論的にどのように組み合わせるかを示し，今後は提案システムが客観的な環境のグローバルな情報に基づいて補間探索を実現できることを確認する．

粒子群最適化 (Particle Swarm Optimization; PSO) は，群の中の固体（粒子）が持つ最良の情報とそのグループの最適値から過去の探索から考慮した確率的最適化手法であり， ケネディ[1] が社会的行動に基づいて開発した並列進化計算技術である． 社会的方法と計算方法の両方を扱う PSO に関する標準的な研究がある [2].

近年，コンピュータサイエンスの発展は，ハードウェアとソフトウェアの有効性が顕著に表れている．その中で大規模問題の最適化の重要性はますます高めっている．ソーシャルネットワークサービスの登場により，ログやパスの問題も大規模になっている．最新のコンピュータでこれらの問題を解決するには時間がかかってしまう．

本研究では数ステップでもっとも最適な解が見つかる新しいハイブリッド動的システムを提供する．そこで連続 PSO アルゴリズムに勾配法を組み込み，定式化を行う．そこで，提案した手法の有効性を示す．その後，提案した PSO の手法を用いて取得したライフログのクラスタリング行い，その結果を示す．

PSO は群をなして移動する生物の行動を模範したアルゴリズムである．群をなす生物をモデル化し，粒子は最適化問題における候補解を示している．PSO は群の中の粒子がもつ最良の情報 (pbest) とその集団の最適値 (gbest) から過去の探索を考慮し，さらにその集団の各粒子の位置および速度を更新することによって計算される (式 4.1，4.2 参照)．以下に PSO の解説を示す (図 4.1 参照).

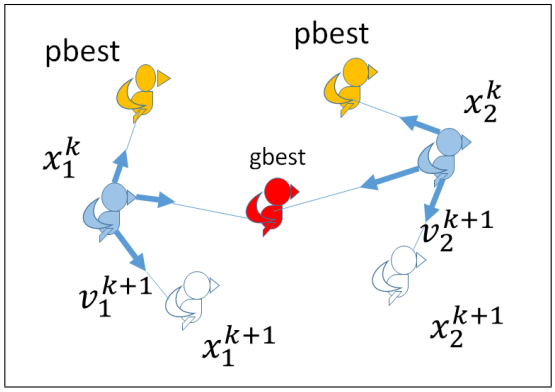


図 4.1: PSO の探索模式図

$$x_d^{k+1} = x_d^k + v_d^{k+1} \tag{4.1}$$

$$v_d^{k+1} = wv_d^k + c_1r_1(x_d^k b - x_d^k) + c_2r_2(x_g^k b - x_d^k) \tag{4.2}$$

ここで，PSO の探索模式図及び速度と位置の更新式より，pbest に向かう $c_1r_1(x_{db}^k - {x_d}^k)$，gbest に向かう $c_2r_2(x_{db}^k - {x_d}^k)$，これまでの進行方向へ向かう wv_d^k の 3 つのベクトルを合成して速度ベクトル ${v_i}^{k+1}$ を決定し，それを元に次に移動する位置 x_d^{k+1} 決定する．

PSO の探索式はランダム要素を含み，同時に最良解情報である pbest と gbest が探索に伴い変化するという時変性を有している [3]. このままの形では理論解析が困難であるので，一つの Particle に着目し，一次元の位置 x と速度 v について考え，さらに pbest と gbest を一つの点に縮約した簡略モデルが提案されている [2]. この簡略モデルは，確定的な線形時不変システムとして表現されており，その安定性を示す.

Particle$i$ に注目すると速度ベクトル v^{k+1} は以下の式のように変形できる (式 4.3,4.4)．ステップ幅 ϕ は二つの一様乱数を足し合わせたものであり，最小値 0，最大値 $C_1 + C_2$，平均 $\frac{C_1+C_2}{2}$ の分布に従う.

$$V^k + 1 = wv^k + \phi(P - x^k) \tag{4.3}$$

$$P = \frac{\phi_1 pbest^k + \phi_2 gbest^k}{\phi_1 + \phi_2} \tag{4.4}$$

ここで，$\phi = \phi_1 + \phi_2$，$\phi_1 = C_1 rand$，$\phi_2 = C_2 rand$，さらに $y^k = p - x^k$ とおくと，式 (4.5) のように表せる．また $\phi = (C_1 + C_2)/2$ と見なすと固有値 λ は式 (4.6) のように表せる．よって λ が 1 を境にシステムの特性が安定・不安定 (収束・発散) に変化することが分かる.

$$\begin{bmatrix} v^k + 1 \\ y^k + 1 \end{bmatrix} = \begin{bmatrix} w & \phi \\ -w & 1 - \phi \end{bmatrix} \begin{bmatrix} v^k \\ y^k \end{bmatrix} \tag{4.5}$$

$$\lambda = \frac{w + 1 - \phi \pm \sqrt{(w + 1 - \phi)^2 - 4w}}{2} \tag{4.6}$$

[文献田中] を元に連続型 PSO アルゴリズム (Continuous Particle Swarm Optimization; CPSO) について述べる．

ベクトル y と $\mathrm{sgn}(y)$ の要素によって与えられる対角要素を持つ対角行列を $\mathrm{diag}[y]$ とする．y の σ 関数を表す．として $\mathrm{sgn}(y) = 1$ if $y > 0$ の場合は, $\mathrm{sgn}(y) = -1$ if $y < 0$.

したがって，正の定数であると仮定すると，最小化のために X の進化を近似することが提案される． また CPSO の安定性解析も議論されている [文献].

状態変数 X，V，X_{db} はベクトルではなく，以前に定義された適切な次元の行列であるため、上記の表記法は標準状態空間表記法ではない． また以下に CPSO の位置と速度の更新式 (式 4.7,4.8) と，アルゴリズムについて示す．問題解決の実行可能領域を考え，行列による連続時間 PSO 動力学を示す (式 4.9,4.10,4.11).

CPSO アルゴリズム

1:X, V とパラメータ α, β, γ と a の初期値を設定する．
2:X_{db}, X_{gb} の初期値を導出する．
3:$\dot{V}$ を計算して,V を更新する．
4:X を更新して X_{db}, X_{gb} を評価する．
5:収束すると仮定した場合は終了しそれ以外の場合は 3 から繰り返す．

$$\dot{X} = V \tag{4.7}$$

$$\dot{V} = -\alpha V + \beta(X_{db} - X) + \gamma(X_{gb}T - X) \tag{4.8}$$

$$\dot{X_{db}} = a(X - X_{db})[I_n + diag[sgn(F(X_{gb} - F(X)))]] \tag{4.9}$$

$$\dot{X_{gb}} = X_{db}Q_j \tag{4.10}$$

$$j = arg \inf_{0<i\leq n}(f(x_{db})) \tag{4.11}$$

オリジナルの PSO アルゴリズムに含まれる恣意性を少なくし，より効率的かつ高精度な探索を実現するために勾配法による速度評価を導入している [文献, 文献]. 運動性素子が自分の置かれた近くの環境を知覚してより適合度の高い空間座標を獲得するために，以下のようなセンサリング・アルゴリズムを搭載する.

勾配によるスケーリングパラメータの導入を行う．粒子が投入された探索空間 (ξ, η) には問題に応じた 目的関数 Q が定義されており，粒子はその最大値か最小値を探索するものとする. 現時間ステップ k における粒子の位置座標を $(\xi k, \eta k)$ とし，その座標における目的関数の値を Q_k として，粒子の移動に伴う目的関数の変化に注目すると次のような目的関数の離散的な勾配 α が得られる.

勾配 α を，$v_i^{k+1} = \beta^k v_i^k$，$\beta^k = \alpha^{k-1/\alpha k}$ と置くことでランドスケープに合わせた調整が可能となる．つまり，最適点が遠いと思うなら早く，近いと感じるならば遅く移動する．よってオリジナル PSO より精密な探索が実施できる.

本節では CPSO に勾配情報の要素を加えた勾配 PSO について解説する．PSO の応用法である CPSO の応用法であり，X, Y の二つの行列に加えて Z を加えかつ，いくつかのパラメータを与えて再急降下法を用いる．以下は Z を表す (式 4.12),

勾配 PSO アルゴリズム

1:X, V とパラメータ α, β, γ と a の初期値を設定する．
2:X_{db}, X_{gb}, X_{μ} から Z の初期値を導出する．
3:$\dot{Z}$ を計算して,Z を更新する．
4:$\dot{V}$ を計算して,V を更新する．
5:X を更新して X_{db}, X_{gb} と Z を評価する．
6:X_{μ} を更新する．
7:収束すると仮定した場合は終了しそれ以外の場合は 3 から繰り返す．

$$Z = \beta(X_{db} - X) + \gamma(X_{gb}T - X) + \gamma(X_{\mu}) \tag{4.12}$$

以下の条件を考慮する (式)．X_0 は初期位置行列である．

$$X = X_0 + \int_0^t V(s)ds \tag{4.13}$$

よって勾配 PSO の更新式を以下に示す (式)．ここでは X, V, X_{db}, X_{gb}, T の次元は簡略化のため省略する．

$$\dot{X} = V \tag{4.14}$$

$$\dot{V} = -\alpha V + Z \tag{4.15}$$

$$\dot{Z} = \beta(\dot{X_{db}} - \dot{X}) + (\dot{X_{gb}}T - \dot{X}) + \delta(\dot{X_{\mu}}) \tag{4.16}$$

$$\dot{X_{db}} = a(X - X_{db})[I_n + diag[sgn(F(X_{gb} - F(X)))]] \tag{4.17}$$

$$\dot{X_{gb}} = X_{db}Q_j, j = arg \inf_{0<i\leq n}(f(x_{db_i})) \tag{4.18}$$

$\alpha, \beta, \gamma, \delta$ などの実数は，PSO と勾配情報を調整するために重み付けするパラメータである． X_{μ} はニューラルネットワークのダイナミクスに由来する新しい行列である． X_{μ} は以下で定義する (図 6 参照)．

$$x_{\mu i} = -C\sum_{i=1} n\frac{\partial f(y_i(t))}{\partial y_i}\frac{\partial \varphi(x(t))}{\partial x_i} \tag{4.19}$$

$$\ddot{x_i} = -\alpha\dot{x_i}(t) + z_i(t) \tag{4.20}$$

$$y_i(t) = \varphi(x_i(t)) \tag{4.21}$$

したがって ${z^k}_i$ は Z のベクトルである．次に，差分法を適用する．理論的な分析の観点から, PSO の $\dot{V}$ と $\dot{x_i}$ は同等のものとみなす. $\beta(\dot{X_{db}} - \dot{X}) + \gamma(\dot{X_{gb}}T - \dot{X})$ は PSO の速度を制御する．$\delta(\dot{X_{\mu}})$ は勾配情報を制御する.

PSO が有するグローバル探索，ニューラルネットワークが持つ局所最急降下法などがある． 連続時間モデルでは，PSO とニューラルネットワークの組み合わせの理論的アルゴリズムが考慮されるが，分散モデルによって数値シミュレーションが行われる． サンプリング時間の設定は，係数 (β, γ, δ) の値によって変化する．したがって，X_{μ} を計算して取得する．

§ 4.2　制約がある場合のPSO

前節まではPSOに勾配情報を加える説明を行った．本節ではそこに制約条件を加える[文献]．連続時間PSOアルゴリズムについて述べる．
PSOの更新式を力学系モデルとみなし，その連続化を試みると，

$$\frac{dx^p(t)}{dt} = c\int_0^t e^{-a(t-\tau)}[F^p(x^p(\tau),\tau) + C(x^p(\tau),\tau)]d\tau \tag{4.22}$$

$$\frac{d^2x^p(t)}{dt^2} + a\frac{dx^p(t)}{dt} = c[F^p(x^p(t),t) + C(x^p(t),t)] \tag{4.23}$$

またそれぞれの関数F^p, Cは以下のようになる．

$$F^p(x^p,t) = c_1(x^p(T^p(t) - x^p)) \tag{4.24}$$

$$C^p(x^p,t) = c_2(x^{Q(t)}(T^o(t) - x^p)) \tag{4.25}$$

また，2階微分方程式で表される連続時間系モデルの状態変数表現を，$u^p(t) = x^p(t)$，$v^p(t) = du^p(t)/dt + au^p(t)$とおいて導入すると，離散時間系に対応した連続系の内部状態表現モデル，

$$du^p(t)/dt = -au^p(t) + v^p(t) \tag{4.26}$$

$$dv^p(t)/dt = c[F^p(u^p(t),t) + C(u^p(t),t))] \tag{4.27}$$

を得ることができる．

次に提案手法であるハイブリッドPSOについて解説する．PSOの応用法である連続時間PSOアルゴリズムの応用法であり，勾配情報を加えることにより，精密な探索を行うことを狙いとしている．
式(4.23,4.24)に勾配情報を加えるとそのモデルは，

$$\frac{dx^p(t)}{dt} = c\int_0^t e^{-a(t-\tau)}[F^p(x^p(\tau),\tau) + C(x^p(\tau),\tau) - \nabla E(x^p(\tau),\tau)]\tau \tag{4.28}$$

$$\frac{d^2x^p(t)}{dt^2} + a\frac{dx^p(t)}{dt} = c[F^p(x^p(t),t) + C(x^p(t),t) - \nabla E(x^p(t),t)] \tag{4.29}$$

またそれぞれの関数は以下のようになる．

$$F^p(x^p,t) = c_1(x^p(T^p(t) - x^p)) \tag{4.30}$$

$$C^p(x^p,t) = c_2(x^{Q(t)}(T^o(t) - x^p)) \tag{4.31}$$

$$\nabla E(x^p,t) = c_3\frac{\partial E(x^p,t)}{\partial x^p} \tag{4.32}$$

解説したままのモデルでは無制約なので，制約条件に対応したモデルである上下限制約連続時間PSOモデルを以下の式に示す．上下限制約付最適化問題

$$minE(x) \tag{4.33}$$

$$subj.top_i \leq x_i \leq q_i, i = 1..., n \tag{4.34}$$

を直接解くために，この上下限制約領域内に問題 (4.34),(4.35) の変数を変換して無制約化した新たな変数空間に無制約 PSO モデルを適用した「変数変換モデル」を導入する．非線形変数変換モデルを作成するために，

$$x_i = f_i(y_i) = \frac{q_i + p_i exp(-y_i)}{1 + exp(-y_i)} \tag{4.35}$$

とおく．この変換式を制約条件付き問題に代入して変数 x を消去すると，

$$minE(f(y)) \tag{4.36}$$

を得ることができる．よって式 (4.30) に対応させると，

$$\frac{d^2y^p(t)}{dt^2} + a\frac{dy^p(t)}{dt} = c[F^p(y^p(t), t) + C(y^p(t), t) - \nabla E(y^p(t), t)] \tag{4.37}$$

またそれぞれの関数は以下のようになる．

$$F^p(y^p, t) = c_1(y^p(T^p(t) - y^p)) \tag{4.38}$$

$$C^p(y^p, t) = c_2(y^{Q(t)}(T^o(t) - y^p)) \tag{4.39}$$

$$\nabla E(y^p, t) = c_3\frac{\partial E(y^p, t)}{\partial y^p} \tag{4.40}$$

次にプログラムへの実装を考えた時に，連続式のままではプログラムに実装することが難しいので，オイラー法を用いて連続式を離散化し非線形変数変換モデルの離散化 PSO を作成．それぞれに対応する式を以下に示す．

$$u^p(k+1) = (1 - a\triangle T)u^p(k) + \triangle T v^p(k) \tag{4.41}$$

$$v^p(k+1) = v^p(k) + c\triangle T[F^p(u^p(k), k) + C(u^p(k), k) - \nabla E(u^p(k), k)] \tag{4.42}$$

$$F^p(k, k) = c_1(u^p(l^p(k)) - u^p(k)) \tag{4.43}$$

$$C^p(k, k) = c_2(u^{Q(k)}(l^o(k)) - u^p(k)) \tag{4.44}$$

$$\nabla E(k, k) = c_3\frac{\partial E(k, t)}{\partial k} \tag{4.45}$$

$$l^p(k) = argmin(E(x^p(l))|l = 0, ..., k) \tag{4.46}$$

$$(Q(k), l^o(k)) = argmin(E(x^q(l))|q = 1, 2, ..., P, l = 0, 1, .., k) \tag{4.47}$$

$$x^p{}_i(k) = f_i(u^p{}_i(k)) = \frac{q_i + p_i exp(-u^p{}_i(k))}{1 + exp(-u^p{}_i(k))}, i = i, ..., n \tag{4.48}$$

提案手法の有効性を示すため評価関数として Rastrigin function[26](図 4.2) と Griewank function[27](図 4.3) を用いて数値実験を行う．これらの評価関数は多峰性ならびに，非常に多くの局所解を持つ．

[26]https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda rastrigin-function
[27]https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda griewank-function

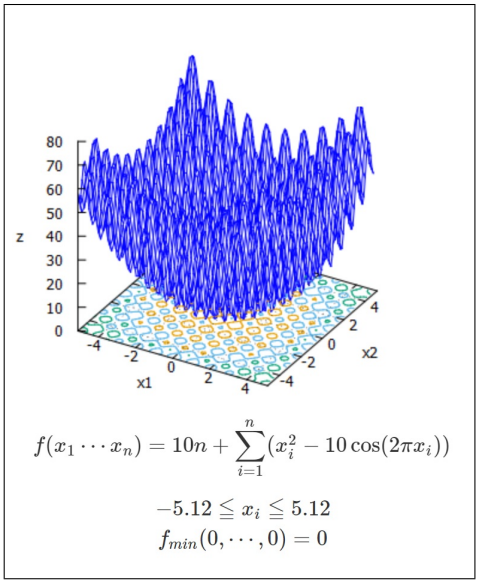


図 4.2: Rastrigin[26]

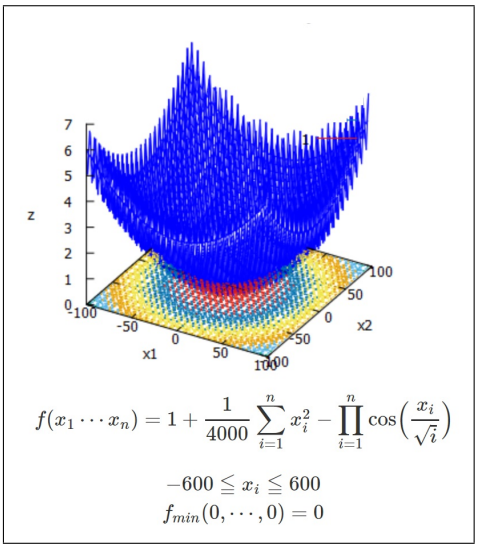


図 4.3: Griewank[27]

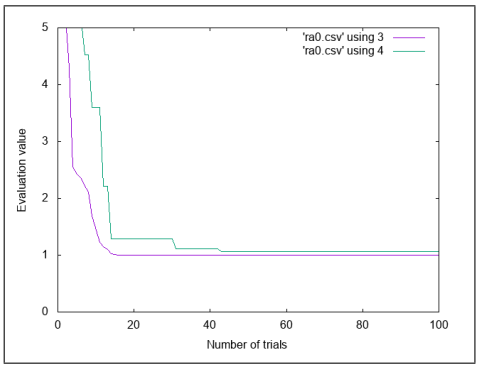


図 4.4: Rastrigin 実行結果

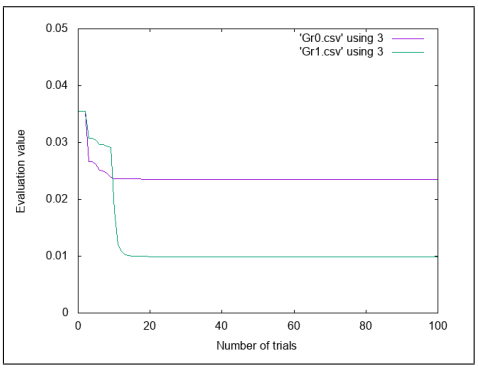


図 4.5: Griewank 実行結果

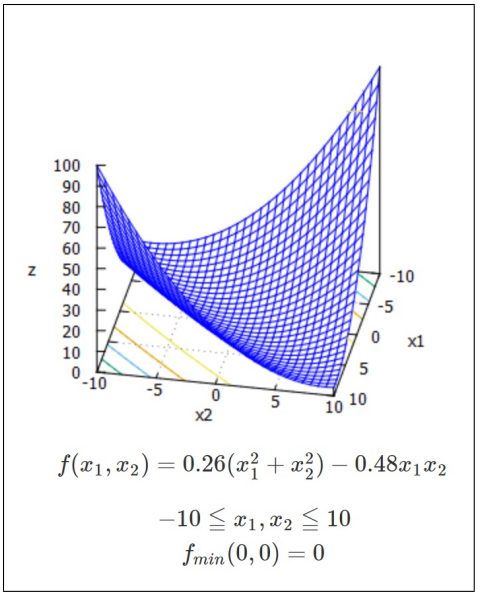
以下に従来法と提案手法を比較した物を示す (図 4,5)．各パラメータの値は N（粒子の数）=10,C=1.0,c_1・c_2=1.4,c_3=0.1,a=1.0,$\triangle T = 0.9$ というように与えた．また試行回数は 100 とした．

加えて評価関数 Matyas function[27](図 4.6) と Booth function[28](4.7) を用いて数値実験を行う．この関数は単峰性関数である．上記の実験と同じ条件で行いその結果を示す (図 4.8, 4.9)．また Booth function の実行時に粒子の数を N=100 に変更した場合の結果 (図 4.10) も示す．

提案手法は多峰性をもつ関数の場合においてあまり良い結果が得られなかった．これは提案手法が勾配法を用いているため多峰性をもつ関数には適していないと思われる. 一方，単峰性をもつ関数の場合においては従来法よりも良い結果を得ることができた．

また発生させる粒子の数を 100 から 10 を比較した場合，従来法では最適値を導出できな

[28]https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda matyas-function
[29]https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda booth-function

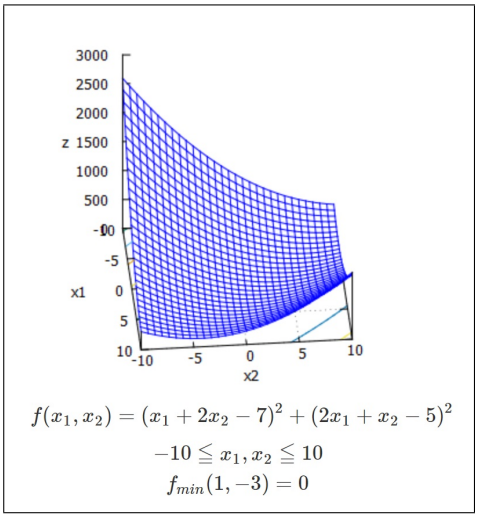


$$f(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$$

$$-10 \leqq x_1, x_2 \leqq 10$$

$$f_{min}(0, 0) = 0$$

図 4.6: Matyas function[28]



$$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

$$-10 \leqq x_1, x_2 \leqq 10$$

$$f_{min}(1, -3) = 0$$

図 4.7: Booth function[29]

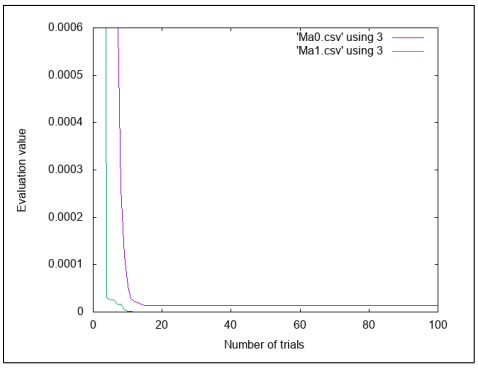


図 4.8: Matyas 実行結果

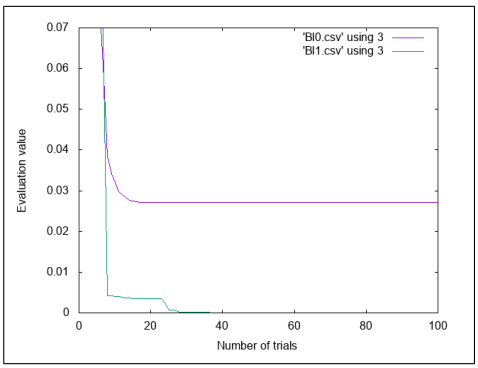


図 4.9: Booth 実行結果

かったが，提案手法では最適値を導出することができた．これにより計算コストの削減が可能となった．

§ 4.3　PSO によるクラスタリング

研究努力により，データクラスタリングを最適化問題と見なすことが可能になった．この見解は，候補クラスタセントロイドの集合を発展させるために PSO アルゴリズムを適用する機会を提供し，それにより手近なデータセットの最適に近い分割を決定する．PSO の重要な利点は，いくつかの候補解を同時に維持し，再結合し，比較することによって局所的な最適条件に対処できることである． 対照的に，シミュレーテッドアニーリングアルゴリズム（Selim and Alsultan，1991）のような局所探索ヒューリスティックは単一の候補解を改良するだけであり，局所最適条件に対処することにおいて悪名高いことが知られている．K 平均法などのアルゴリズムで使用される決定論的局所検索は，常に検索の開始位置

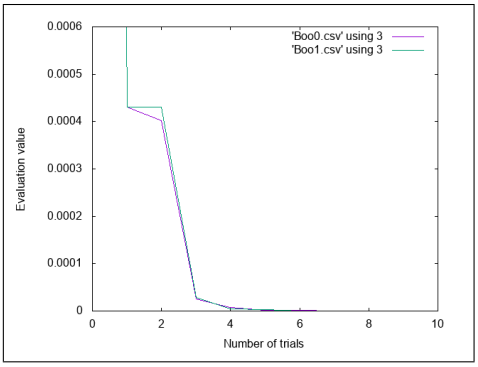


図 4.10: Booth 実行結果 N=100 の場合

から最も近い局所最適条件に収束する．

PSO ベースのクラスタリングアルゴリズムは，Omran らによって最初に導入された．（Omran ら，2002）に記載されている．Omran らの結果（Omran et al.、2002、Omran et al。、2005a）は，PSO ベースの方法が K-means，FCM，その他いくつかの最先端のクラスタリングアルゴリズムより優れていることを示した．Omran らの方法ではクラスタリングアルゴリズムの性能を判断するために，量子化誤差に基づく適応度を使用した． 量子化誤差は次のように定義される．

$$J_e = c_3 \frac{\sum_{i=1}^{K} \sum_{\forall X_j \in C_i} d(X_j, Vi)/n_i}{K} \tag{4.49}$$

ここで，C_i は i 番目のクラスタ中心，n_i は i 番目のクラスタに属するデータ点の数である．PSO アルゴリズムの各粒子は，次のように K 個のクラスタ重心の可能なセットを表す．

$$\overrightarrow{Z}_i(t) = \overrightarrow{V}_{i,1}, \overrightarrow{V}_{i,2}, ..., \overrightarrow{V}_{i,K} \tag{4.50}$$

ここで，V_i，p は，i 番目の粒子の p 番目のクラスター重心ベクトルを表す． 各粒子の品質は，次の適合度関数によって測定される．

$$f(Z_i, M_i) = w_1 \bar{d}_{max} + w_2(R_{max} - d_{min}(Z_i)) + w_3 J_e \tag{4.51}$$

上記の式で，R_{max} はデータセット内の最大の特徴量であり，M_i は i 番目の粒子のクラスターへのパターンの割り当てを表す行列である． 各要素 m_i，k は，パターン X_p が i 番目の粒子のクラスタ C_k に属するかどうかを示す． ユーザー定義定数 w_1，w_2，および w_3 は，異なる副目的からの寄与を評価するために使用される． 加えて，

$$\bar{d}_{max} = \underbrace{max}_{K \in 1,2....,K} \left(\underbrace{\sum}_{\forall X_p \in C_i, K} d(X_p, V_{i,k})/n_{i,k} \right) \tag{4.52}$$

そして，

$$d_{min}(Z_i) = \underbrace{min}_{\in p,q,p \neq q} \left(d(V_{i,p}, V_{i,q}) \right) \tag{4.53}$$

上記の式は任意のペアのクラスタ間の最小ユークリッド距離である．上記において，ni, k は，粒子 i のクラスタ C_i, k に属するパターンの数である．適応度関数は多目的最適化問題であり，これはクラスタ内距離を最小化し，クラスタ間分離を最大化し，量子化誤差を減少させる．PSO クラスタリングアルゴリズムは，以下に示す．

PSO クラスタリングアルゴリズム

1 : K 個のランダムクラスター中心を使って各粒子を初期化する．
2 : for 繰り返し回数 =1 から最大繰り返し回数まで do
3 : for 全ての粒子に対して do
4 : for データセット内のすべてのパターン X_p に対して行う do
5 : すべてのクラスター重心を使って Xp のユークリッド距離を計算する
6 : Xp に最も近い重心を持つクラスターに X_p を割り当てる
7 : end
8 : フィットネス関数 $f\,(Z_i, M_i)$ を計算する
9 : end
10 : 各パーティクルの個人的および世界的に最良の位置を見つける
11 : PSO の式を更新する．
12 : end

Van der Merwe と Engelbrecht は，一般的なデータセットをクラスタリングするための kmeans アルゴリズムである（van der Merwe と Engelbrecht，2003）. 群の単一粒子は，k-means アルゴリズムの結果で初期化される． スウォームの残りの部分はランダムに初期化される． 2003 年に，Xiao らは PSO と自己組織化マップ（SOM）との相乗作用に基づく新しいアプローチを用いて遺伝子発現データをクラスタリングした（Xiao ら，2003）． 彼らは，酵母およびラット肝細胞の遺伝子発現データに対してハイブリッド SOM-PSO アルゴリズムを適用することによって有望な結果を得た．Paterlini and Krink（Paterlini and Krink，2006）は，分割に対する代表的な点評価アプローチについて，Kmeans，GA（Holland，1975，Goldberg，1975）PSO および微分進化（DE）（Storn and Price，1997）の性能を比較した． クラスタリング 結果は PSO と DE が K 平均アルゴリズムより優れていることを示した．

Ｃｕｉ ｅｔ ａｌ.（Cui and Potok，2005）は，テキスト文書を分類するための PSO ベースのハイブリッドアルゴリズムを提案した．彼らは PSO，K 平均，そして 4 つの異なるテキスト文書データセット上のハイブリッド PSO クラスタリングアルゴリズム．結果はハイブリッド PSO アルゴリズムがよりコンパクトを生成できることを示した．クラスタリングは、K-means アルゴリズムよりも短時間で行われる．

§ 4.4 提案手法のアルゴリズム

第5章

結論ならびに今後の課題

本研究の目的は，多くの人に広く受け入れられるライフログとして，個人情報保護に着目し，手間がかからず自動的にライフログデータの取得を行い，取得したデータから類似性やイベント性を考察できることである．開発したライフログデータ取得アプリケーションを使用したビッグデータ構築・データ解析を行い，行動パターンの類似性・イベント検出を行った．

結論として，個人情報保護に着目したライフログデータ取得アプリケーションの開発ができ，多変量解析を用いることでライフログの可視化を行い行動パターンの類似性やイベント性を視覚的に検出するという目標は達成できた．特に，SOM の解析結果より，同じ行動でも視界に写る物体の違いから行動の類似性やイベント性を検出できた．同じ行動でも使用する場所や物体の変化によって別行動として認識させることができるため，ライフログデータに位置情報を付加できると考えられる．よって，個人情報保護に着目し取得したライフログデータから類似性やイベント性を検出できたと考える．本研究の研究成果は，テキストによるライフログデータ取得，解析を行い新たなビジネスプランの検討やユーザー自身の生活の見直しなどに使用できるため，より高度なアプリケーション開発を目指す開発者，研究者の方々の参考になれば幸いである．解明できた点は必ずしも多くはないが，若干なりとも寄与できたと思われる．

今後の課題として，開発したアプリケーションの改善点を上げる．開発したアプリケーションは自動的にライフログデータを取得する点が利点として挙げられるが，一方で客観的なライフログデータしか取得できないという弱点もある．ユーザーが興味を持った瞬間や，データを取得したい瞬間のライフログデータは現状のアプリケーションには含まれていないためである．この弱点に対し，ユーザーが取得したいタイミングでライフログデータを取得する方法をアプリケーションに組み込む必要がある．組み込むため，取得したいタイミングを MOVERIO^{TM} に伝える方法の検討も必要となる．

謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学電子・情報工学科の奥原浩之教授に深甚な謝意を表します．最後になりましたが，多大な協力をして頂いた研究室の同輩諸氏に感謝致します．

2018 年 2 月

福嶋 瑞希

参考文献

[1] 中西泰人，辻貴孝，大山実，箱崎勝也 “Context Aware Messaging Service：位置情報とスケジュール情報を用いたコミュニケーションシステムの構築及び運用実験” 情報処理学会論文誌，Vol. 42，No. 7，pp.1847–1857，2001

[2] 相澤清晴, “ライフログ”, 映像情報メディア学会誌, Vol. 63, No. 4, pp. 445–448, 2009.

[3] 三井紀子，酒井豊子，中島利誠 “運動中の衣服下気候と着心地に及ぼす繊維の影響” 日生気誌，Vol. 23，No. 1，pp.35–42，1986

[4] 柳平雅俊，安士光男 “運転状態推定技術の開発— 心拍解析による眠気状態の検出—” PIONEER R & D，Vol. 14，No. 3，pp.17–27，2003

[5] 前中一介 “絆創膏型人体活動モニタリングシステム” 人工臓器，Vol. 42，No. 1，pp.17–27，2013

[6] 芳竹宣裕, 伊藤愼, “ユビキタス環境が生み出す大量情報 「ライフログ」 の活用と実装技術”, NEC 技報, Vol. 62, No. 4, p. 77, 2009.

[7] 新保史生, “ライフログの定義と法的責任 個人の行動履歴を営利目的で利用することの妥当性”, 情報管理, Vol. 53, No. 6, pp. 295–310, 2010.

[8] 角田宏貴, Hiroki SUMIDA, “ライフログ分析による行動特徴抽出及びイベント検出”, 法政大学大学院紀要（情報科学研究科編）, Vol. 9, pp. 119–124, 2014.

[9] 矢野裕司, 横井健, 橋山智訓, “行動辞書を利用した Twitter からの行動抽出”, 情報科学技術フォーラム講演論文集, Vol. 11, No. 4, pp. 51–56, 2012.

[10] 緒方広明, “日本語学習を支援するユビキタス学習環境に関する研究”, http://www.taf.or.jp/files/items/542/File/P212.pdf, 閲覧日 2018,1,30.

[11] 啓之田中, “位置情報の規律のあり方：スマートフォン時代の利便性とプライバシー”, 人間社会研究, Vol. 11, pp. 75–85, 2014.

[12] 北村圭吾, 山崎俊彦, 相澤清晴, “食事ログの取得と処理－画像処理による食事記録－”, 映像情報メディア学会誌, Vol. 63, No. 3, pp. 376–379, 2009.

[13] 株式会社ＮＴＴデータ経営研究所, “日本語学習を支援するユビキタス学習環境に関する研究”, http://www.keieiken.co.jp/aboutus/newsrelease/161122/, 閲覧日 2018,2,5.

[14] 入江英嗣, 森田光貴, 岩崎央, 千竃航平, 放地宏佳, 小木真人, 樫原裕大, 芝星帆, 眞島一貴, 努吉永, “AirTarget：光学シースルー方式 HMD とマーカレス画像認識による高可搬性実世界志向インタフェース”, 情報処理学会論文誌, Vol. 55, No. 4, pp. 1415–1427, 2014.

[15] 川上晃平, “スマートグラスを利用した授業支援システムの開発”, 2017.

[16] 倉田陽平, 真田風, 鈴木祥平, 石川博, “Flickr と Google Cloud Vision API によりテーマ別観光マップを作る試み”, http://db-event. jpn.org/deim2017/papers/321.pdf, 閲覧日 2018,1,4.

[17] 大雄治, 吉川眞, 田中一成, “ソーシャルメディアを活用した景観の分析と評価”, 日本都市計画学会関西支部研究発表会講演概要集, Vol. 15, pp. 13–16, 2017.

[18] 小林亜令, 岩本健嗣, 西山智, “釈迦：携帯電話を用いたユーザ移動状態推定・共有方式-モバイルコンピューティング, モバイルアプリケーション, ユビキタス通信, モバイルマルチメディア通信-”, 電子情報通信学会技術研究報告. MoMuC, モバイルマルチメディア通信, Vol. 108, No. 44, pp. 115–120, 2008.

[19] 寺田努, “ウェアラブルセンサを用いた行動認識技術の現状と課題”, コンピュータ ソフトウェア, Vol. 28, No. 2, pp. 43–54, 2011.

[20] 貴志一樹, 山崎俊彦, 相澤清晴, “机上行動のライフログのための行動認識”, 映像情報メディア学会年次大会講演予稿集, Vol. 2014, , 2014.

[21] 前川卓也, 柳沢豊, 岸野泰恵, 石黒勝彦, 亀井剛次, 櫻井保志, 岡留剛, “ウェアラブルセンサによるモノを用いた行動の認識について”, Technical Report 57, 研究報告ユビキタスコンピューティングシステム（UBI）, 2010.

[22] 樋口耕一, “テキスト型データの計量的分析：2 つのアプローチの峻別と統合”, 理論と方法, Vol. 19, No. 1, pp. 101–115, 2004.

[23] 佐野香織, 李在鎬, “KH Coder で何ができるか：日本語習得・日本語教育研究利用への示唆”, 言語文化と日本語教育, Vol. 33, pp. 94–95, 2007.

[24] “KH Coder を用いた研究事例のリスト”, http://khc.sourceforge.net/bib.html, 閲覧日 2018,1,7.

[25] 二宮隆次, 小野浩幸, 高橋幸司, 野田博行, “新聞記事を基にしたテキストマイニング手法による産学官連携活動分析”, 科学・技術研究, Vol. 5, No. 1, pp. 93–104, 2016.

[26] “クラスター分析の手法（階層クラスター分析） — データ分析基礎知識”, https://www.albert2005.co.jp/knowledge/data_mining/cluster/hierarchical_clustering, 閲覧日 2018,1,24.

[27] “階層的クラスター分析について”, http://koichi.nihon.to/cgi-bin/bbs_khn/khcf.cgi?list=&no=977&mode=allread&page=0, 閲覧日 2018,1,24.

[28] 吉原一紘, 徳高平蔵, “クラスター分析の概要”, *Journal of Surface Analysis*, Vol. 21, No. 1, pp. 10–17, 2014.

[29] 李美龍, 田中恒也, 成田吉弘, “画像を用いた製品の「飽き」に関する感性評価: －デザインの視覚的要素を中心に－”, 日本感性工学会論文誌, Vol. 11, No. 3, pp. 407–417, 2012.

[30] 小峯敦・下平裕之, “ベヴァリッジ『自由社会における完全雇用』のケインズ的要素-テキストマイニングを加味した量的・質的分析-”, 2017.

[31] 齋藤堯幸, “多次元尺度構成法”, 計測と制御, Vol. 22, No. 1, pp. 126–131, 1983.

[32] Joseph B Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”, *Psychometrika*, Vol. 29, No. 1, pp. 1–27, 1964.

[33] “Jaccard 係数の計算式と特徴（1）”, https://www.slideshare.net/khcoder/jaccard1, 閲覧日 2018,2,3.

[34] 中山慶一郎, “< 研究ノート > 対応分析によるデータ解析”, 関西学院大学社会学部紀要, No. 108, pp. 133–145, 2009.

[35] 田中京子, “KH Coder と R を用いたネットワーク分析”, 久留米大学コンピュータジャーナル, Vol. 28, pp. 37–52, 2014.

[36] “共起ネットワークにおける中心性の解釈について”, http://www.koichi.nihon.to/cgi-bin/bbs_khn/khcf.cgi?no=2493&mode=allread#2496, 閲覧日 2018,2,2.

[37] 横田尚己, 山田圭二郎, “熊本地震のつぶやきに見る感情極性値の時空間解析”, 都市計画論文集, Vol. 52, No. 3, pp. 1081–1087, 2017.

[38] 増田正, Masuda Tadashi, 高崎経済大学地域政策学部, “地方議会の会議録に関するテキストマイニング分析 : 高崎市議会を事例として”, 地域政策研究 = Studies of regional policy, Vol. 15, No. 1, pp. 17–31, 2012.

[39] T. KOHONEN, “Self-organized formation of topologically correct feature map”, *Biol. Cybern.*, Vol. 43, pp. 59–69, 1982.

[40] 岡晋之介, “自己組織化マップを用いた気象要素の分類と予測”, http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/oka/oka.html, 閲覧日 2018,1,7.

[41] “自己組織化特徴マップ（SOM）”, http://www.sist.ac.jp/ kanakubo/research/neuro/selforganizingmap.html, 閲覧日 2018,1,31.

[42] “KH Coder 掲示板”, http://koichi.nihon.to/cgi-bin/bbs_khn/khcf.cgi?&no=3457&reno=3454&oya=3454&mode=msgview, 閲覧日 2018,1,20.

[43] 勝治宏基, 米澤拓郎, 中澤仁, 高汐一紀, 徳田英幸ほか, “Synchrometer: ライフログを利用した日常行動における他者との類似度生成”, 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2013, No. 17, pp. 1–7, 2013.

[44] “R-Tips”, http://cse.naro.affrc.go.jp/takezawa/r-tips.pdf, 閲覧日 2018,1,20.

付録

A. 1　ライフログデータ取得アプリケーションのソースコード

ライフログデータ取得アプリケーションのソースコード A.1 をしめす.

ソースコード A. 1: app.cs

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine.UI;
5  using System.IO;
6  using System;
7  using System.Text;
8  using System.Linq;
9
10 public class app : MonoBehaviour
11 {
12   private float captureIntervalSeconds = 50.0f;
13   private float captureIntervalSeconds2 = 5.0f;
14   public Text gtext;
15   Dictionary<string, string> headers;
16   private int Width = 1280;
17   private int Height = 720;
18   private int FPS = 30;
19   private WebCamTexture webcamTexture;
20   private Color32[] color32;
21   string responseData;
22   private string reportFileName2 = "long_report.txt";
23   public bool addDateTime = true;
24
25   void Start ()
26   {
27     Screen.sleepTimeout = SleepTimeout.NeverSleep;
28     StartCoroutine ("Sample");
29   }
30
31   public IEnumerator Sample ()
32   {
33     WebCamDevice[] devices = WebCamTexture.devices;
34     WebCamDevice userCameraDevice = WebCamTexture.devices [0];
35     webcamTexture = new WebCamTexture (userCameraDevice.name, Width, Height,
           FPS);
36     webcamTexture.Play ();
37     Debug.Log ("webcamTexture");
38
39     yield return new WaitForSeconds (captureIntervalSeconds2);
40     color32 = webcamTexture.GetPixels32 ();
41     Texture2D texture = new Texture2D (webcamTexture.width, webcamTexture.height
           );
42     texture.SetPixels32 (color32);
43     texture.Apply ();
44     byte[] jpg = texture.EncodeToJPG ();
45     string VISIONKEY = "API KEY";
```

```
46      var uri = "https://westus.api.cognitive.microsoft.com/vision/v1.0/
            describe";
47
48
49      var headers = new Dictionary<string, string> () {
50        { "Ocp-Apim-Subscription-Key", VISIONKEY },
51        { "Content-Type", "application/octet-stream" }
52      };
53
54      WWW www = new WWW (uri, jpg, headers);
55      yield return www;
56      responseData = www.text;
57      gtext.text = responseData;
58        DateTime dt = DateTime.Now;
59        string text2 = dt.ToString ("[yyyy-MM-dd HH:mm:ss]") + responseData.ToString
              () + "\n";
60        string outfile2 = reportFileName2;
61
62        if (addDateTime) {
63          string file2 = Path.GetFileNameWithoutExtension (reportFileName2);
64          string ext2 = Path.GetExtension (reportFileName2);
65          outfile2 = file2 + "_" + dt.ToString ("yyyyMMdd") + ext2;
66        }
67
68        SaveText (text2, Path.Combine (Application.persistentDataPath, outfile2));
69        color32 = null;
70      StartCoroutine ("StopRunTimeTemp");
71    }
72
73    public IEnumerator StopRunTimeTemp ()
74    {
75      webcamTexture.Stop ();
76      yield return new WaitForSeconds (captureIntervalSeconds);
77      StartCoroutine ("Sample");
78    }
79
80    public static bool SaveText (string text, string path)
81    {
82      try {
83        using (StreamWriter writer = new StreamWriter (path, true)) {
84          writer.Write (text);
85          writer.Flush ();
86          writer.Close ();
87        }
88      } catch (Exception e) {
89        Debug.Log (e.Message);
90        return false;
91      }
92      return true;
93    }
94  }
```

A. 2　時系列SOMを作成するソースコード

時系列 SOM を作成するソースコード A.2 をしめす.

ソースコード A. 2: som.r

```
d <- NULL
d <- matrix(
c(1,1,1,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,
2,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,
3,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,
4,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
5,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
6,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
7,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
8,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
9,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
10,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
11,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
12,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
13,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
14,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
15,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
16,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
17,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,
 0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,
18,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
19,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
 0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,
20,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
 1,1,1,0,0,1,0,0,0,1,0,0,0,0,0,0,
21,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,
22,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,
23,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
24,2,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
25,1,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
26,2,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
27,1,0,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
28,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
29,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
30,1,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
31,1,1,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
32,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
33,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
34,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
35,1,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
36,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
37,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
38,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
39,2,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
40,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
41,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
42,2,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
43,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
44,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
45,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
46,2,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
47,1,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
48,2,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
49,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
50,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
51,2,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
52,2,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
53,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
54,0,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
55,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
56,2,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
57,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
58,1,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
59,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
60,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
```

61 60,0,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
62 61,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
63 62,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
64 63,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
65 64,2,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
66 65,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
67 66,2,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
68 67,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
69 68,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
70 69,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
71 70,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
72 71,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
73 72,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
74 73,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
75 74,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
76 75,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
77 76,2,0,0,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
78 77,2,0,0,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
79 78,1,1,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
80 79,2,0,0,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
81 80,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
82 81,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
83 82,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
84 83,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
85 84,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
86 85,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
87 86,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
88 87,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
89 88,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
90 89,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
91 90,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
92 91,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
93 92,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
94 93,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
95 94,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
96 95,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
97 96,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
98 97,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
99 98,1,2,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
100 99,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
101 100,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
102 101,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
103 102,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
104 103,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
105 104,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
106 105,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
107 106,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
108 107,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
109 108,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
110 109,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
111 110,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
112 111,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
113 112,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
114 113,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
115 114,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
116 115,2,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
117 116,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
118 117,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
119 118,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
120 119,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
121 120,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,

```
121,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
122,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
123,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
124,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
125,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
126,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
127,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
128,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
129,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
130,2,1,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
131,2,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
132,2,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
133,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
134,1,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
135,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
136,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
137,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
138,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
139,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
140,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
141,2,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
142,1,2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
143,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
144,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
145,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
146,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
147,2,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
148,2,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
149,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
150,1,0,0,2,0,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
151,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
152,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
153,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
154,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
155,1,0,0,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
156,2,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
157,1,0,0,2,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
158,2,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
159,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
160,2,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
161,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
162,1,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
163,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
164,1,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
165,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
166,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
167,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
168,2,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
169,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
170,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
171,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
172,0,0,0,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
173,0,0,0,2,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
174,1,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,
175,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
176,0,0,0,2,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,2,0,0,0,0,
177,1,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
178,2,0,0,2,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
179,2,0,0,2,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
180,2,0,0,2,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
181,2,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
```

183 182,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,1,0,0,
184 183,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,2,0,0,
185 184,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
186 185,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,
187 186,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
188 187,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
189 188,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
190 189,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,0,0,0,0,0,
191 190,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
192 191,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
193 192,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
194 193,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
195 194,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
196 195,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
197 196,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
198 197,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
199 198,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
200 199,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
201 200,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
202 201,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
203 202,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
204 203,0,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
205 204,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
206 205,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
207 206,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
208 207,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
209 208,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
210 209,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
211 210,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
212 211,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
213 212,1,0,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
214 213,2,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
215 214,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
216 215,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
217 216,0,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
218 217,1,0,2,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
219 218,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
220 219,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
221 220,0,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
222 221,1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
223 222,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,
224 223,2,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
225 224,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,
226 225,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
227 226,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
228 227,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
229 228,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
230 229,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
231 230,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
232 231,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
233 232,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
234 233,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
235 234,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
236 235,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
237 236,1,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
238 237,1,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
239 238,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
240 239,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
241 240,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
242 241,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
243 242,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,

244 243,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,
245 244,2,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
246 245,0,2,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
247 246,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
248 247,2,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
249 248,0,1,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
250 249,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
251 250,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
252 251,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
253 252,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
254 253,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
255 254,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
256 255,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
257 256,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
258 257,0,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
259 258,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,1,1,0,0,0,
260 259,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
261 260,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
262 261,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,0,1,0,0,0,
263 262,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
264 263,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
265 264,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
266 265,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
267 266,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
268 267,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
269 268,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,0,0,0,0,0,
270 269,0,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
271 270,1,0,1,0,0,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
272 271,0,2,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
273 272,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
274 273,1,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
275 274,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
276 275,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
277 276,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
278 277,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
279 278,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
280 279,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
281 280,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
282 281,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
283 282,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
284 283,0,0,2,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,
285 284,0,1,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
286 285,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
287 286,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
288 287,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
289 288,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
290 289,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
291 290,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
292 291,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
293 292,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
294 293,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
295 294,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
296 295,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
297 296,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
298 297,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
299 298,0,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
300 299,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,0,0,0,
301 300,0,1,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
302 301,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
303 302,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
304 303,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,1,

305 304,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,
306 305,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,2,0,0,
0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,
307 306,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,2,0,0,
0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,
308 307,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0,0,
309 308,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
2,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
310 309,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,
311 310,1,2,2,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
312 311,1,2,1,0,0,0,0,0,0,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
313 312,0,1,0,0,0,0,0,0,2,0,0,2,0,0,1,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
314 313,0,1,2,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
315 314,0,2,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
316 315,2,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
317 316,0,2,0,0,0,0,0,0,0,0,0,0,2,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
318 317,0,2,0,0,0,0,0,0,0,0,0,0,2,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
319 318,0,1,0,0,0,0,0,0,0,0,0,2,0,0,2,0,0,0,0,0,0,1,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
320 319,0,2,0,0,0,0,0,0,1,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
321 320,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
322 321,0,2,0,0,0,0,0,0,0,0,1,0,2,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
323 322,0,1,0,0,0,0,0,0,2,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
324 323,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
325 324,0,2,0,0,0,1,0,0,2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
326 325,0,2,0,0,0,0,0,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
327 326,2,2,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
328 327,2,2,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
329 328,0,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,1,0,
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
330 329,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
331 330,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
332 331,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
333 332,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
334 333,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
335 334,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
336 335,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
337 336,0,1,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
338 337,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
339 338,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
340 339,0,1,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
341 340,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
342 341,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
343 342,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
344 343,0,2,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
345 344,0,1,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
346 345,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
347 346,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
348 347,0,0,2,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,
349 348,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,
350 349,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,
351 350,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,
352 351,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,
353 352,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
354 353,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
355 354,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
356 355,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
357 356,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
358 357,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
359 358,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
360 359,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
361 360,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
362 361,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
363 362,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
364 363,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
365 364,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,

```
365,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
366,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,
367,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
368,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
369,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
370,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
371,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
372,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
373,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
374,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
375,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
376,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
377,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
378,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
379,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
380,1,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0), byro
  w=T, nrow=380, ncol=41 )
d <- d[,-1]
colnames(d) <- c("desk","table","l
  aptop","monitor","desktop","keyboa
  rd","screen","shot","person","clos
  e","electronics","food","flyer","c
  ounter","plate","snow","bottle","c
  ar","cat","microwave","oven","cake
  ","kitchen","bird","building","cab
  inet","door","man","mirror","stand
  ing","water","window","computer","
  indoor","open","black","laptop","w
  hite","outdoor","small")
doc_length_mtr <- matrix( c(
  70,18,63,18,63,18,63,18,68,19,63,18,63,
  18,63,18,68,19,69,19,65,19,69,19,57,18,
  62,18,54,18,62,18,60,19,60,18,55,18,69,
  20,57,19,36,14,69,19,72,19,61,18,78,20,
  59,18,64,18,64,18,61,18,58,18,66,18,69,
  19,69,19,70,19,69,19,69,19,69,19,71,19,
  72,19,69,19,71,19,69,19,69,19,69,19,78,
  20,61,18,78,20,64,18,69,19,72,19,72,19,
  69,19,62,18,69,19,72,19,64,18,56,18,64,
  18,64,18,66,18,69,19,67,19,71,19,59,18,
  72,19,69,19,69,19,65,19,75,20,61,17,69,
  19,70,19,70,19,69,19,79,20,79,20,62,18,
  79,20,70,19,69,19,67,19,69,19,69,19,58,
  18,69,19,69,19,69,19,69,19,58,18,69,19,
  69,19,70,19,70,19,69,19,69,19,64,18,70,
  19,66,20,70,19,65,18,66,19,66,19,71,19,
  71,19,71,19,70,19,71,19,70,19,70,19,70,
  19,70,19,70,19,65,18,66,18,67,20,67,20,
  67,20,67,20,71,19,67,20,71,19,71,19,71,
  19,71,19,71,19,67,20,67,20,71,19,77,20,
  71,19,70,19,65,18,75,18,65,18,65,18,65,
  18,65,18,65,18,65,18,65,18,67,19,66,19,
  70,19,70,19,70,19,70,19,74,19,68,18,86,
  21,68,18,68,18,68,18,68,18,68,18,76,19,
  75,18,71,18,70,18,74,19,75,19,79,19,77,
  19,75,19,75,19,68,18,77,19,76,19,77,19,
  77,19,75,19,72,18,79,18,77,19,75,19,79,
  18,70,18,83,20,83,20,83,20,71,18,59,18,
  56,19,61,17,63,19,64,21,53,17,54,17,72,
  21,45,17,75,20,75,20,75,20,75,20,75,20,
  75,20,75,20,75,20,75,20,76,20,76,20,76,
  20,70,20,76,20,73,19,73,19,73,19,73,19,
  73,19,73,19,73,19,72,19,62,18,62,18,70,
  19,60,18,65,19,54,17,66,19,59,18,56,18,
  59,19,58,20,59,19,61,20,61,20,64,19,77,
  20,66,20,69,19,66,20,61,18,66,18,68,19,
  66,19,58,19,53,17,48,17,50,17,65,21,60,
  21,65,19,58,19,85,24,89,24,72,20,85,24,
  62,19,69,20,69,20,69,20,72,20,69,20,69,
  20,73,20,69,20,71,20,70,20,73,20,73,20,
  72,20,73,20,56,16,56,16,75,20,56,16,60,
  16,70,20,88,24,71,19,86,24,76,20,88,24,
  76,20,76,20,76,20,73,20,71,20,64,19,75,
  20,75,20,76,20,77,20,62,19,75,20,69,20,
  69,20,56,16,60,16,57,16,57,16,57,16,70,
  20,70,20,70,20,72,20,70,20,89,24,76,20,
  72,20,57,16,72,20,56,18,70,20,54,18,51,
  18,85,24,54,18,49,18,72,21,74,23,64,19,
  77,23,59,19,56,18,56,19,55,19,63,21,83,
  26,62,19,59,20,62,19,76,23,79,23,68,20,
  87,25,87,25,55,18,81,22,78,22,78,22,81,
  22,72,19,61,17,88,24,90,24,70,18,88,24,
  90,24,88,24,87,24,87,24,88,24,59,17,70,
  18,72,19,74,20,53,18,57,18,61,20,72,19,
  54,18,57,19,57,19,56,18,54,18,54,18,54,
  18,54,18,54,18,57,19,54,18,54,18,54,18,
  54,18,54,18,57,19,57,19,57,19,57,19,57,
  19,57,19,57,19,57,19,57,19,57,19,57,19,
  57,19,57,19,57,19), ncol=2, byrow=T)
colnames(doc_length_mtr) <- c("leng
  th_c", "length_w")

d <- t(d)
n_nodes <- 20
rlen1 <- 1000
rlen2 <- 200000

d <- t(d)
if (exists("doc_length_mtr")){
  leng <- as.numeric(doc_length_mt
    r[,2])
  leng[leng ==0] <- 1
  d <- d / leng
  d <- d * 1000
}
```

```
d <- subset(d, rowSums(d) > 0)
d <- scale(d)
d <- t(d)
d <- t(d)
rownames(d) <- 1:nrow(d)
# SOM
library(som)
somm <- som(
  d,
  n_nodes,
  n_nodes,
  topol="hexa",
  rlen=c(rlen1,rlen2)
)

word_labs <- rownames(d)
n_words <- length(word_labs)
color_universal_design <- 1
cex <- 1
text_font <- 2
if_cls <- 1
n_cls <- 9
if_plothex <- 1

row2coods <- NULL
eve <- 0
for (i in 0:(n_nodes - 1)){
  for (h in 0:(n_nodes - 1)){
    row2coods <- c(row2coods, h + eve, i)
  }
  if (eve == 0){
    eve <- 0.5
  } else {
    eve <- 0
  }
}
row2coods <- matrix( row2coods, byrow=T, ncol=2 )


if ( if_cls == 1 ){
  library( RColorBrewer )

  if (
       ( as.numeric( R.Version()$major ) >= 3 )
    && ( as.numeric( R.Version()$minor ) >= 1.0)
  ){ # >= R 3.1.0
    hcl <- hclust( dist(somm$code,method="euclidean"), method="ward.D2" )
  } else { # <= R 3.0
    hcl <- hclust( dist(somm$code,method="euclidean")^2, method="ward" )
  }

  colors <- NULL
  if (n_cls <= 9){
    pastel <- brewer.pal(9, "Pastel1")
    pastel[6] = "gray91"
    pastel[9] = "#F5F5DC" # FAF3C8 F7F1C6 EEE8AA F0E68C
    colors <- pastel[cutree(hcl,k=n_cls)]
  }
  if (n_cls > 9) {

    library(colorspace)
    new_col <- order( runif(n_cls) )
    colors <-
      rainbow_hcl(n_cls, start=20, end=340, l=92, c=20)[
      #terrain_hcl(n_cls, c = c(35, 5), l = c(85, 95), power = c(0.5,1))[
        new_col[cutree(hcl,k=n_cls)]
      ]
  }
} else {
  colors <- rep("gray90", n_nodes^2)
}
labcd <- NULL

plot_mode <- "color"
par(mai=c(0,0,0,0), mar=c(0,0,0,0), omi=c(0,0,0,0), oma =c(0,0,0,0) )

plot(
  NULL,NULL,
  xlim=c(0,n_nodes-0.5),
  ylim=c(0,n_nodes-1),
  axes=F,
  frame.plot=F
)

if (if_plothex == 1){
  a <- 0.333333333333
} else {
  a <- 0.5
}
b <- 1-a

color_pte <- "gray70"
cls_lwd <- 2

if ( plot_mode == "gray"){
  color_act <- rep("white",n_nodes^2)
  if_points <- 1
  w_lwd <- 1
  cls_lwd <- 2.25
  color_cls <- "gray35"
  color_line <- "gray50"
  color_pte <- "gray40"
  color_ptf <- "gray85"
}
if ( plot_mode == "color" ) {
  color_act <- colors
```

```
    color_line <- "white"
    if_points <- 1
    w_lwd <- 1
    if (n_cls > 9) {
      color_cls <- "gray45"
    } else {
      color_cls <- "gray60"
    }
    color_ptf <- "white"
  }
  if ( plot_mode == "freq" ){
    color_act <- somm$code.sum$nobs;
    if (max(color_act) == 1){
      color_act <- color_act * 3 + 1;
    } else {
      color_act <- color_act - min(colo
      r_act)
      color_act <- round( color_act / m
      ax(color_act) * 6 ) + 1
      #color_act[color_act==7] <- 6
    }
    color_seed <- brewer.pal(6,"GnBu")
    color_seed <- c("white", color_see
    d)
    color_act <- color_seed[color_act]

    color_line <- "gray70"
    if_points <- 0
    w_lwd <- 1
    color_cls <- "gray45"
    color_ptf <- "white"
  }
  if ( plot_mode == "umat" ){
    dist_u <- NULL
    dist_m <- as.matrix( dist(somm$co
    de, method="euclid") )
    for (i in 0:(n_nodes - 1)){
      for (h in 0:(n_nodes - 1)){
        cu <- NULL
        n <- 0

        if (h != n_nodes -1){
          cu <- c(
            cu,
            dist_m[
              h + i * n_nodes + 1,
              h + 1 + i * n_nodes + 1
            ]
          )
        }

        if (h != 0){
          cu <- c(
            cu,
            dist_m[
              h + i * n_nodes + 1,
              h - 1 + i * n_nodes + 1
            ]
          )
        }

        if (i != n_nodes - 1){
          if (h %% 2 == 0){
            cu <- c(
              cu,
              dist_m[
                h + i * n_nodes + 1,
                h + ( i + 1 ) * n_nodes
                + 1
              ]
            )
          } else {
            if (h != n_nodes -1){
              cu <- c(
                cu,
                dist_m[
                  h + i * n_nodes + 1,
                  h + 1 + ( i + 1 ) * n_
                  nodes + 1
                ]
              )
            }
          }
        }

        if (i != 0){
          if (h %% 2 == 0){
            cu <- c(
              cu,
              dist_m[
                h + i * n_nodes + 1,
                h + ( i - 1 ) * n_nodes
                + 1
              ]
            )
          } else {
            if (h != n_nodes -1){
              cu <- c(
                cu,
                dist_m[
                  h + i * n_nodes + 1,
                  h + 1 + ( i - 1 ) * n_
                  nodes + 1
                ]
              )
            }
          }
        }

        if (i != n_nodes - 1){
          if (h %% 2 == 0){
            if (h != 0){
              cu <- c(
                cu,
                dist_m[
                  h + i * n_nodes + 1,
                  h - 1 + ( i + 1 ) * n_
                  nodes + 1
                ]
              )
            }
          } else {
```

```
            cu <- c(
              cu,
              dist_m[
                h + i * n_nodes + 1,
                h + ( i + 1 ) * n_nodes
                + 1
              ]
            )
          }
        }

        if (i != 0){
          if (h %% 2 == 0){
            if (h != 0){
              cu <- c(
                cu,
                dist_m[
                  h + i * n_nodes + 1,
                  h - 1 + ( i - 1 ) * n_
                  nodes + 1
                ]
              )
            }
          } else {
            cu <- c(
              cu,
              dist_m[
                h + i * n_nodes + 1,
                h + ( i - 1 ) * n_nodes
                + 1
              ]
            )
          }
        }
        dist_u <- c(dist_u, median(cu) )
      }
    }

    print( summary(dist_u) )

    dist_u <- dist_u - min(dist_u)
    dist_u <- round( dist_u / max(dist_
    u) * 100 ) + 1

    if (color_universal_design == 0){
      color_act <- cm.colors(101)[dist_
      u]
      color_line <- "gray70"
      color_cls <- "gray45"
    } else {
      library(RColorBrewer)
      if (T){
        col_seed <- brewer.pal(9, "GnBu
        ")
        myPalette <- colorRampPalett
        e( col_seed )
        color_act <- myPalette(101)[dis
        t_u]
        color_act <- adjustcolor(color_a
        ct, alpha=0.8)
        color_line <- "white"
        color_cls <- "gray30"
      } else {
        col_seed <- rev(brewer.pal(9, "
        RdYlBu"))
        myPalette <- colorRampPalett
        e( col_seed )
        color_act <- myPalette(101)[dis
        t_u]
        color_act <- adjustcolor(color_a
        ct, alpha=0.8)
        color_line <- "gray50"
        color_cls <- "gray25"
      }
    }
    if_points <- 1
    w_lwd <- 1
    color_ptf <- "white"
}


for (i in 1:n_nodes^2){
    x <- row2coods[i,1]
    y <- row2coods[i,2]

    polygon(
      x=c( x + 0.5, x + 0.5, x, x - 0.5,
      x - 0.5, x ),
      y=c( y + a, y - a, y - b, y - a, y
       + a, y + b ),
      col=color_act[i],
      border="white",
      lty=0,
    )
}


for (i in 0:(n_nodes - 1)){
    for (h in 0:(n_nodes - 2)){
      if ( colors[h + i * n_nodes + 1]
      == colors[h + i * n_nodes + 2] ){
        x <- h
        y <- i
        if ( y %% 2 == 1 ){
          x <- x + 0.5
        }

        segments(
          x + 0.5, y + a,
          x + 0.5, y - a,
          col=color_line,
          lwd=w_lwd,
        )
      }
    }
}

for (i in 0:(n_nodes - 1)){
    for (h in c(-1, n_nodes-1) ){
      x <- h
      y <- i
      if ( y %% 2 == 1 ){
```

```
725         x <- x + 0.5
726       }
727       segments(
728         x + 0.5, y + a,
729         x + 0.5, y - a,
730         col=color_line,
731         lwd=w_lwd,
732       )
733     }
734     if ( y %% 2 == 0 ){
735       segments(
736         -0.5, y + a,
737         0 , y + 1 - a,
738         col=color_line,
739         lwd=w_lwd,
740       )
741       if ( y != 0){
742         segments(
743           -0.5, y - a,
744            0 , y - 1 + a,
745           col=color_line,
746           lwd=w_lwd,
747         )
748       }
749     } else {
750       if ( y != n_nodes - 1){
751         segments(
752           n_nodes - 0.5, y + 1 - a,
753           n_nodes , y + a,
754           col=color_line,
755           lwd=w_lwd,
756         )
757       }
758       segments(
759         n_nodes - 0.5, y - 1 + a,
760         n_nodes , y - a,
761         col=color_line,
762         lwd=w_lwd,
763       )
764     }
765   }
766
767   for (i in 0:(n_nodes - 2)){
768     for (h in 0:(n_nodes - 1)){
769       if (i %% 2 == 1){
770         chk <- 1
771       } else {
772         chk <- 0
773       }
774
775       if (
776             is.na(colors[h + i * n_nodes
                + 1]) == 1
777         || is.na(colors[h + chk + (i+1) *
              n_nodes + 1]) == 1
778         || h + chk == n_nodes
779       ){
780         next
781       }
782
783       if (
784             colors[h + i * n_nodes + 1]
785         == colors[h + chk + (i+1) * n_
              nodes + 1]
786       ){
787         x <- h
788         y <- i
789         if ( y %% 2 == 1 ){
790           x <- x + 0.5
791         }
792
793         segments(
794           x, y + b,
795           x + 0.5, y + a,
796           col=color_line,
797           lwd=w_lwd,
798         )
799       }
800     }
801   }
802
803   for (i in 0:(n_nodes - 2)){
804     for (h in 0:(n_nodes - 1)){
805       if (i %% 2 == 0){
806         chk <- 1
807       } else {
808         chk <- 0
809       }
810       if (
811             is.na(colors[h + i * n_nodes
                + 1]) == 1
812         || is.na(colors[h - chk + (i+1) *
              n_nodes + 1]) == 1
813         || h - chk < 0
814       ){
815         next
816       }
817
818       if (
819             colors[h + i * n_nodes + 1]
820         == colors[h - chk + (i+1) * n_
              nodes + 1]
821       ){
822         x <- h
823         y <- i
824         if ( y %% 2 == 1 ){
825           x <- x + 0.5
826         }
827
828         segments(
829           x, y + b,
830           x - 0.5, y + a,
831           col=color_line,
832           lwd=w_lwd,
833         )
834       }
835     }
836   }
837
838   for (i in 0:(n_nodes - 1)){
839     for (h in 0:(n_nodes - 2)){
840       if ( colors[h + i * n_nodes + 1] !=
```

```
      colors[h + i * n_nodes + 2] ){
841     x <- h
842     y <- i
843     if ( y %% 2 == 1 ){
844       x <- x + 0.5
845     }
846
847     segments(
848       x + 0.5, y + a,
849       x + 0.5, y - a,
850       col=color_cls,
851       lwd=cls_lwd,
852     )
853   }
854 }
855
856 for (i in 0:(n_nodes - 2)){
857   for (h in 0:(n_nodes - 1)){
858     if (i %% 2 == 1){
859       chk <- 1
860     } else {
861       chk <- 0
862     }
863
864     if (
865         is.na(colors[h + i * n_nodes
            + 1]) == 1
866       || is.na(colors[h + chk + (i+1) *
          n_nodes + 1]) == 1
867       || h + chk == n_nodes
868     ){
869       next
870     }
871
872     if (
873         colors[h + i * n_nodes + 1]
874       != colors[h + chk + (i+1) * n_n
            odes + 1]
875     ){
876       x <- h
877       y <- i
878       if ( y %% 2 == 1 ){
879         x <- x + 0.5
880       }
881
882       segments(
883         x, y + b,
884         x + 0.5, y + a,
885         col=color_cls,
886         lwd=cls_lwd,
887       )
888     }
889   }
890 }
891
892 for (i in 0:(n_nodes - 2)){
893   for (h in 0:(n_nodes - 1)){
894     if (i %% 2 == 0){
895       chk <- 1
896     } else {
897       chk <- 0
898     }
899
900     if (
901         is.na(colors[h + i * n_nodes
            + 1]) == 1
902       || is.na(colors[h - chk + (i+1) *
          n_nodes + 1]) == 1
903       || h - chk < 0
904     ){
905       next
906     }
907
908     if (
909         colors[h + i * n_nodes + 1]
910       != colors[h - chk + (i+1) * n_n
            odes + 1]
911     ){
912       x <- h
913       y <- i
914       if ( y %% 2 == 1 ){
915         x <- x + 0.5
916       }
917
918       segments(
919         x, y + b,
920         x - 0.5, y + a,
921         col=color_cls,
922         lwd=cls_lwd,
923       )
924     }
925   }
926 }
927
928 points <- NULL
929 sf <- 0.35
930 a <- a * sf;
931 b <- b * sf;
932 c <- 0.5 * sf;
933 for (i in 1:nrow(somm$visual)){
934   x <- somm$visual[i,1]
935   y <- somm$visual[i,2]
936   if ( y %% 2 == 1 ){
937     x <- x + 0.5
938   }
939   points <- c(points, x, y)
940 }
941 points <- matrix( points, byrow=T, n
    col=2 )
942
943 if( if_points == 1 ){
944   if (F){
945     for (i in 1:nrow(points)){
946       x <- points[i,1]
947       y <- points[i,2]
948
949       polygon(
950         x=c( x + c, x + c, x, x - c, x
            - c, x ),
951         y=c( y + a, y - a, y - b, y -
            a, y + a, y + b ),
```

```
          col=color_ptf,
          border=color_pte,
          lty=1,
        )
      }
    } else {
      symbols(
        points[,1],
        points[,2],
        squares=rep(0.35,length(point
        s[,1])),
        #circles=rep(0.2,length(point
        s[,1])),
        fg="gray70",
        bg=color_ptf,
        inches=F,
        add=T,
      )
    }
  }

  library(maptools)
  if (is.null(labcd) == 1){
    labcd <- pointLabel(
      x=points[,1],
      y=points[,2],
      labels=word_labs,
      doPlot=F,
      cex=cex,
      offset=0
    )


    xorg <- points[,1]
    yorg <- points[,2]
    #cex <- 1

    if ( length(xorg) < 300 ) {
      library(wordcloud)

  filename <- tempfile()
  writeLines("wordlayout <- functio
    n (x, y, words, cex = 1, rotat
    e90 = FALSE, xlim = c(-Inf,
      Inf), ylim = c(-Inf, Inf), tste
      p = 0.1, rstep = 0.1, ...)
  {
    tails <- \"g|j|p|q|y\"
    n <- length(words)
    sdx <- sd(x, na.rm = TRUE)
    sdy <- sd(y, na.rm = TRUE)
    iterations <- 0
    if (sdx == 0)
      sdx <- 1
    if (sdy == 0)
      sdy <- 1
    if (length(cex) == 1)
      cex <- rep(cex, n)
    if (length(rotate90) == 1)
      rotate90 <- rep(rotate90, n)
    boxes <- list()
    for (i in 1:length(words)) {
      rotWord <- rotate90[i]
      r <- 0
      theta <- runif(1, 0, 2 * pi)
      x1 <- xo <- x[i]
      y1 <- yo <- y[i]
      wid <- strwidth(words[i], ce
      x = cex[i], ...)
      ht <- strheight(words[i], ce
      x = cex[i], ...)
      if (grepl(tails, words[i]))
        ht <- ht + ht * 0.2
      if (rotWord) {
        tmp <- ht
        ht <- wid
        wid <- tmp
      }
      isOverlaped <- TRUE
      while (isOverlaped) {
        if (!.overlap(x1 - 0.5 * wi
        d, y1 - 0.5 * ht, wid,
          ht, boxes) && x1 - 0.5 *
            wid > xlim[1] && y1 -
          0.5 * ht > ylim[1] && x1
            + 0.5 * wid < xlim[2] &
          &
          y1 + 0.5 * ht < ylim[2])
          {
          boxes[[length(boxes) +
          1]] <- c(x1 - 0.5 * wid,
            y1 - 0.5 * ht, wid, ht)
          isOverlaped <- FALSE
        }
        else {
          theta <- theta + tstep
          r <- r + rstep * tstep/(2
            * pi)
          x1 <- xo + sdx * r * cos(
          theta)
          y1 <- yo + sdy * r * sin(
          theta)
          iterations <- iterations
          + 1
          if (iterations > 500000){
            boxes[[length(boxes) +
            1]] <- c(x1 - 0.5 * wi
            d,
            y1 - 0.5 * ht, wid, ht)
            isOverlaped = FALSE
          }
        }
      }
    }
  print( paste(\"iterations: \", ite
  rations) )
  result <- do.call(rbind, boxe
  s)
  colnames(result) <- c(\"x\", \
  "y\", \"width\", \"ht\")
  rownames(result) <- words
  result
```

```
}
", filename)
insertSource(filename, package="wordcloud", force=FALSE)
nc <- wordlayout(
        labcd$x,
        labcd$y,
        word_labs,
        cex=cex * 1.25,
        xlim=c( par( "usr" )[1], par( "usr" )[2] ),
        ylim=c( par( "usr" )[3], par( "usr" )[4] )
      )

      xlen <- par("usr")[2] - par("usr")[1]
      ylen <- par("usr")[4] - par("usr")[3]

      segs <- NULL
      for (i in 1:length(word_labs) ){
        x <- ( nc[i,1] + .5 * nc[i,3] - labcd$x[i] ) / xlen
        y <- ( nc[i,2] + .5 * nc[i,4] - labcd$y[i] ) / ylen
        dst <- sqrt( x^2 + y^2 )
        if ( dst > 0.05 ){
          segs <- rbind(
            segs,
            c(
              nc[i,1] + .5 * nc[i,3], nc[i,2] + .5 * nc[i,4],
              xorg[i], yorg[i]
            )
          )
        }
      }

      xorg <- labcd$x
      yorg <- labcd$y
      labcd$x <- nc[,1] + .5 * nc[,3]
      labcd$y <- nc[,2] + .5 * nc[,4]
    }


}

if ( exists("segs") ){
  if ( is.null(segs) == F){
    for (i in 1:nrow(segs) ){
      segments(
        segs[i,1],segs[i,2],segs[i,3],segs[i,4],
        col="gray60",
        lwd=1
      )
    }
  }
}

text(
  labcd$x,
  labcd$y,
  labels=word_labs,
  cex=cex,
  offset=0,
  font=text_font
)

if ( exists("out_coord") == F ) {
  out_coord <- cbind(
    labcd$x / (n_nodes-0.5),
    labcd$y / (n_nodes-1)
  )
}

points1<-head(points,n=190)
par(new=T)
plot(points1[,1],points1[,2],type="c",col="red")

points2<-tail(points,n=190)
par(new=T)
plot(points2[,1],points2[,2],type="c",col="blue")
```