迷路で寄り道してみる

藤野稔寛

１．入り口

私は、２０年以上前に盲学校に勤めていたことがあり、そのとき、図形を点訳するソフト「エーデル」を開発した。エーデルは発展を続け、現在では点図や図入り点訳本を作成するために無くてはならないツールとして普及している。そして、点図というものが広く提供されるようになったが、わかりやすい点図の作成方法を標準化することと共に、点字使用者自身の点図触読能力をいかに育成するかということもひとつの課題となっている。そこで、私は、点図による迷路を作成・提供して、点字使用者が迷路で遊ぶことにより点図に慣れ親しんでもらうことができないかと考えた。

実は、エーデルを開発した初期の頃、点図迷路を自動作成する機能をおまけとして付けていた。が、十分満足のいくものでなかったので、今回はこの完成度を上げることを考えた。具体的には、道幅だけでなく、全体のサイズも変更できるようにするなどである。こうして改良の構想を練っているうち、迷路の難易度を設定できるようにしたいと思うようになった。しかし、迷路の難易度はいったい何によって決まるのだろうか？

この疑問について考えるためにも、とにかく迷路自動作成ソフト、名付けて「迷作」を作り、シミュレーションしてみることにした。以下、迷路で寄り道してみた顛末を述べる。

２．出発

ここで扱う迷路は、次のようなものである。

(1) 全体の形は長方形であり、縦に m個、横に n個の格子点が並んでいる。
(2) 隣り合う格子点をつなぐように道があり、左上角と右下角の格子点を入り口・出口とする最短の経路（以後、「解である道」、または、「解となる道」という。）がひとつだけある。
(3) すべての道はひとつに繋がっており、すべての格子点へ行くことができる。

なお、このような迷路を描く場合、壁を描く方法と、道自体を描く方法とがある。これらは本質的に同等であるが、これを点図迷路にして触読するときには大きな違いがある。点図迷路の場合、おそらく、壁ではなく道自体を凸点の列で描く方が、他の点図との共通性があり、適切なのであろう。今回制作した「迷作」では、相互に変換できるようにした。（図１、図２）

迷路を自動生成するアルゴリズムとしてすぐ思い付くのは、迷路を作る壁を描く以下のような方法である。まず、長方形の壁を描き、入り口と出口を開ける。次に、壁の任意の位置から内側へ壁を適当に描く。このとき、その付加した壁が別の壁に着くことはないようにする。後はこのような壁の付加を繰り返すが、付加する位置は、初めの長方形の壁だけでなく、後から付加された壁の上でもよい。これで壁による迷路が出来上がる。しかし、これをパソコンで実現するとき、壁を付加する場所や向きはランダムに決められるので、出来上がる迷路がどのようなものになるか、まして、その難易度をコントロールすることはできそうにない。

迷路の難易度が何によって決まるのかはしばらく措くとして、難易度を調節するためには、とにかく解となる道をまず描くことが必要であるように思えた。では、長方形の形に並んだ格

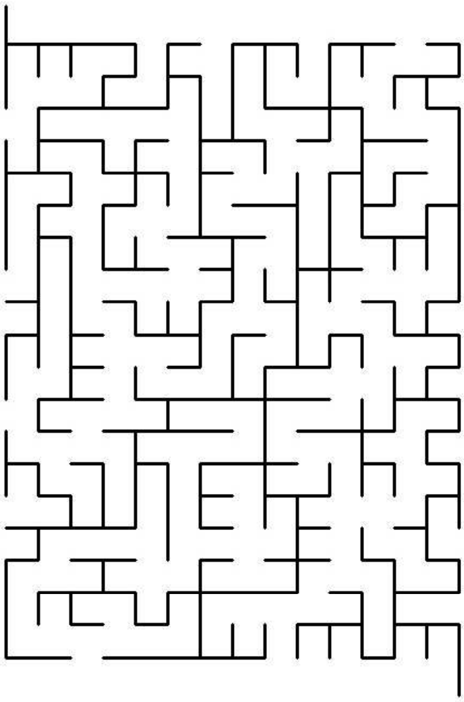


図１　道による迷路

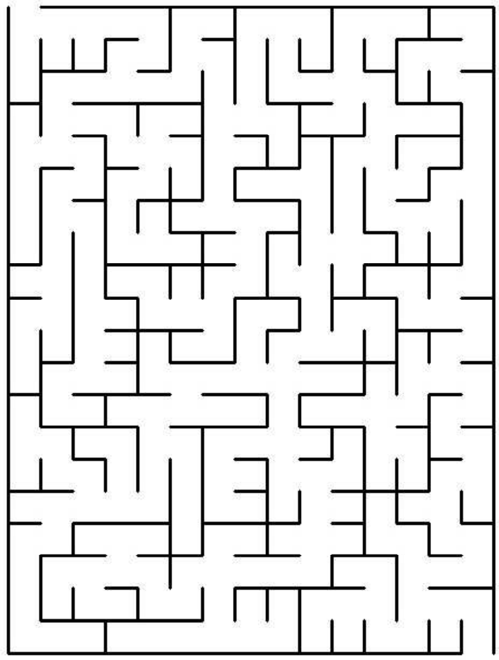


図２　壁による迷路

子点の隣どうしを結びながら左上角から右下角まで一本の折れ線を描くアルゴリズムとして、どのようなものが考えられるであろうか。それは、左上角からランダムに方向を変えながら次々と格子点をつなげていけば良いという訳にはいかない。必ず右下角の終点に着けるようにするためには、途中で行き詰まってしまうことがないようにする必要があるからである。そこで、私は、まず左上角と右下角を最短距離で結ぶ道（図３）を描き、これを適当に曲げ伸ばすこと

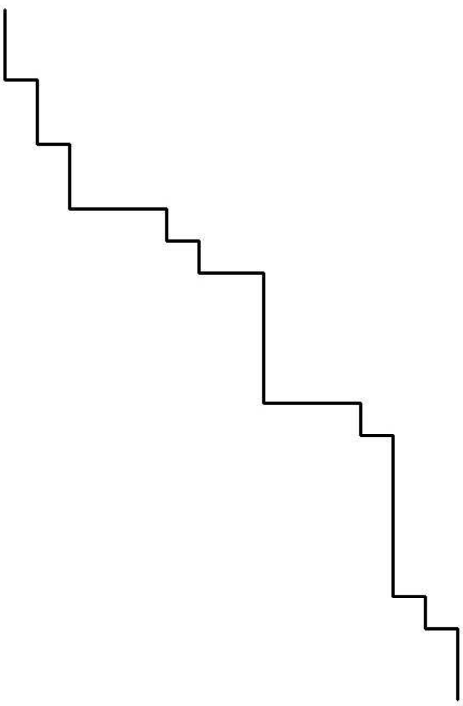


図３　最短経路

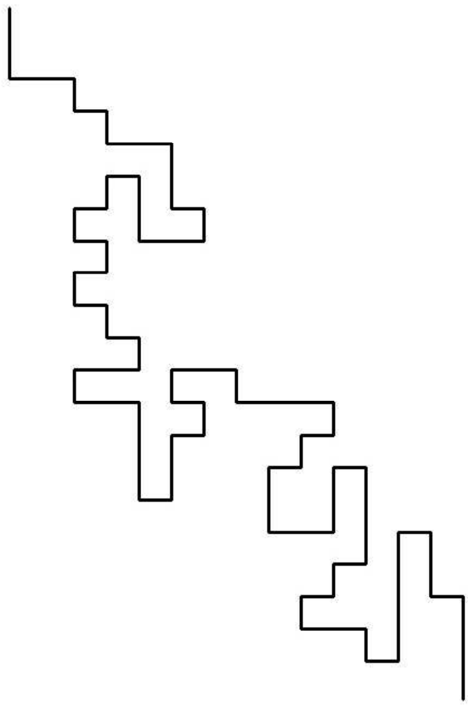


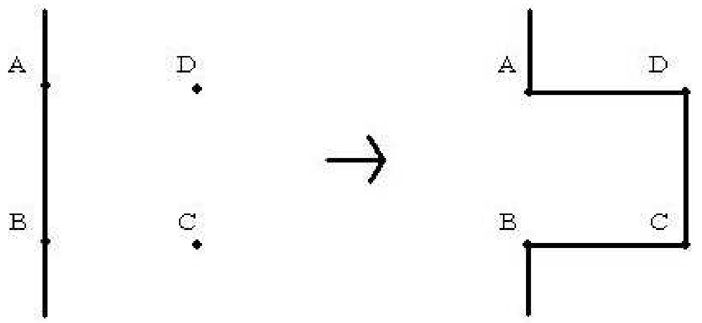
図４　解となる道

で解となる道（図４）を決めれば良いと考えた。これに適当に枝道を付加することで、壁ではなく道自体で描く迷路が出来上がる。

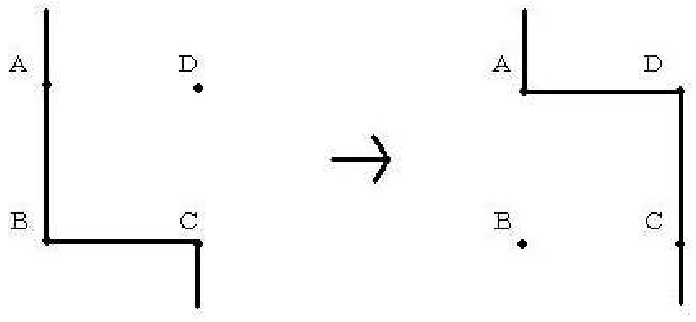
こう考えたのは､解である道がくねくねと長くなっていれば難しい迷路になるのではないか、従って、解である道の長さで難易度が調節できるのではないか、という感じが何となくあったからである。しかし、これが誤りであることは明らかである。なぜなら、解となる道を長くすればするほど枝道は少なくなり、極端な場合、枝道が無くなってしまえば、解である道は非常にくねくねとしているかもしれないが、迷う必要のない一本道になるからである。迷わせるためには道の分岐が必ず必要である。

とにかくこのアルゴリズムを実現してみることにした。まず、左上角から右下角へ最短距離の道を描くことは簡単で、左上角から、進める範囲で右、または、下方向にのみ格子点を繋げていけば良い。このとき、縦に ｍ個、横に ｎ個の格子点を繋ぐので、右へ進むか、下へ進むかの確率を ｎ：ｍ の比に配分するのが良いであろう。もちろん、右端に着いた場合は次は下へ、下端に着いた場合は次は右へ進むしかない。次に、これを曲げ伸ばすにはどうすればいいだろうか。私は次のふたつの操作をおこなうことにした。

操作１ ： 隣接する４つの格子点からなる正方形 ＡＢＣＤ において、２つの点、例えば Ａ と Ｂ が解となる道上の隣り合った格子点であり、Ｃ と Ｄ が道上にない場合に､道 －Ａ－Ｂ－ を道 －Ａ－Ｄ－Ｃ－Ｂ－ にすることで道を長くする。道の「長さ」を道が通過する格子点の数で定義すると、この操作によって道の長さは ２ だけ増える。



操作２ ： 隣接する４つの格子点からなる正方形 ＡＢＣＤ において、道が、例えば －Ａ－Ｂ－Ｃ－ と繋がっており、Ｄ が道上にない場合、道 －Ａ－Ｂ－Ｃ－ を道 －Ａ－Ｄ－Ｃ－ に換える。



３．寄り道

ここでちょっと寄り道し、上記の２つの操作をランダムに繰り返して、初め最短距離だった道を最長の道まで変形できるかどうか試してみた。「最長の道」というのは、解である道が長方形いっぱいに伸びきっているものである。先述した、迷う必要のない一本道である。(図５)ただし、m、n 共に偶数の場合は、格子点の総数 m n が偶数であるのに対して、最短距離 m＋n－１ が奇数であり、従って、これを ２ ずつ伸ばした道の長さも奇数であるため、上記のふたつの操作で作った最長の道では格子点をひとつ残すことになる。(図６)

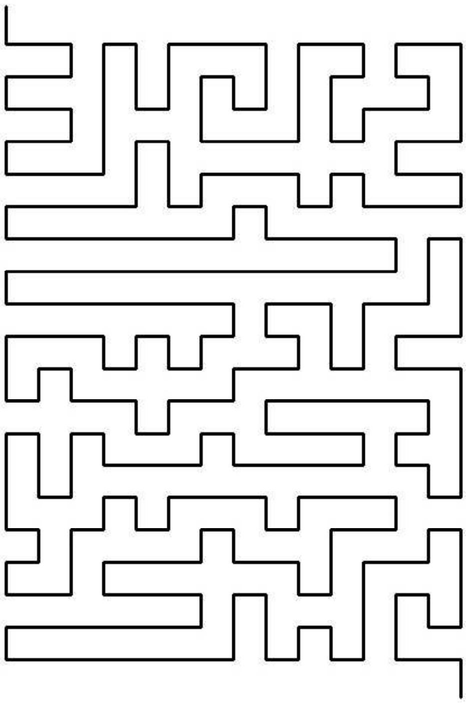


図５　最長の道（m＝３０，n＝１５）
（すべての格子点を通過している）

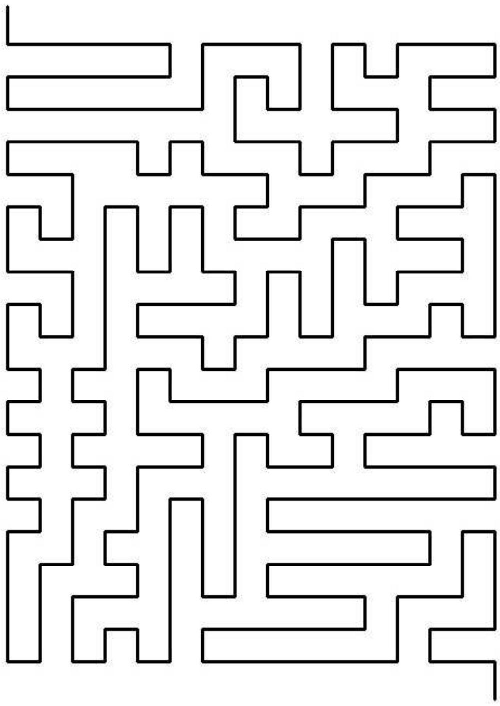


図６　最長の道（m＝３０，n＝１６）
（通過していない格子点が１ヶ所ある）

それにしても、上記の操作に依存しない、次の命題は真だろうか？

『縦に m個、横に n個の格子点を並べる。m、n が共に偶数のとき、左上角から右下角まで、すべての格子点を通る一本道は描けない。ただし、道は水平、または、垂直方向に隣り合う格子点を繋いでいくものとする。』

これは証明できる。格子点の間隔を １ とする。左上角の格子点から道を描き進むことを考えると、道順がどうであれ、右方向には結果的に n－１ だけ進まなければならない。途中で左方向へ戻ることが a 回あるとすると、その分右方向へ戻らなければならないので、道の水平方向の距離の合計は n－１＋２a である。同様に、垂直方向の距離の合計は m－１＋２b と表せるので、道の距離はこれらを足して m＋n＋２（a＋b－１） となる。よって、この道が通過する格子点の数は m＋n＋２（a＋b－１）＋１ であり、奇数である。しかし、格子点の総数 m n は偶数であるので、すべての格子点を通る一本道は描けない。

また、縦、横の少なくとも一方が奇数の場合にすべての格子点を通過する一本道が描けることも、簡単に示すことができる。そのような道の具体的な描き方があるからである。縦方向の

格子点数　ｍ　が奇数であるとき、まず右方向に端まで進み、下へひとつ折れて、今度は左へ端まで進む。こうした横の大きな蛇行を繰り返せば、すべての格子点を通過して右下角へ到達できる。横方向の格子点数　ｎ　が奇数の場合、同様に縦方向の大きな蛇行を繰り返せばよい。

パソコンによるシミュレーションでは、最初に最短距離の道を描き、これに操作１、操作２をランダムに繰り返して最長の道を描こうとした。これでいろいろな経路が描けるのであるが、格子点を２つ以上残すこともあった。その様子を見ていると、いろいろな疑問が生じてきた。とくに、２つの操作をランダムにではなく考えながら実行するとどうなるのだろうか？

［１］最短距離で結ぶ道順は　${}_{m-1+n-1}C_{m-1}$　通りあるが、では、最長の道順は何通りあるのだろうか？

［２］いずれの最短距離の道順からスタートしても、どの最長距離の道順へも、上記の２つの操作だけで必ず伸長・変形できるのだろうか？　ただし、いずれの最短距離の道順も、操作２　によって、最も外側のみを通る　Ｌ字型の道に変形できるので、スタート時点での最短距離の道順は、この　Ｌ字型に限ることができる。

［３］もし、２．が不可の場合、操作１　と操作２　だけでは道の長さは伸びる一方なので、操作１　の逆の操作も加えるとどうなのだろうか？

すべて、私には解けない疑問である。寄り道は得てして行き止まるものなのであろうか？

４．本道

さて、寄り道が長くなり過ぎた。難易度を調節して迷路を作るという出口を目指して本道に戻ろう。先述のようにして、解となる道の長さを調節できるようになったので、後はこれに枝道を付ければ迷路が完成する。このとき、もちろん伸ばした先をすでに描かれている道に付けてはならない。この単位長さの枝道の付加をランダムに繰り返しても良いが、伸ばした先からさらに伸ばせるときは伸ばすこともあるようにして、枝道を少し長くするように工夫した。どの方向に伸ばすかは偶然性に任せた。

こうして迷路を完成させられるようになったので、いよいよ迷路の難易度について改めて考えた。

初め、解となる道が長くてくねくねしている方が難しい、しかし、長すぎると枝道が少なくなってしまうので、逆に易しくなるだろうと考え、解となる道の長さを　ｘ　とすると、難しさは　ｘ（ｍｎ－ｘ）　で与えられるのではないか、と考えた。しかし、ｘ　と　ｍｎ－ｘ　の積ではなく、和　ｘ＋ｍｎ－ｘ＝ｍｎ　かもしれず、そうすると、結局、格子点の総数、つまり、全体のサイズで決まるということになる。これはある意味当然で、頷けない答えではないが、難易度が全体のサイズだけで決まるものでないことは、最長の道の存在を考えれば明らかである。

次に、分岐点の数が多いと難しくなるのではないか、と考えた。そこで、解となる道の長さ　ｘ　を変えながら、ひとつの　ｘ　について１０回ずつ迷路を作り、その分岐点の数を数えるシミュレーションをおこなった。その結果を表１に示す。三叉路は１個、四叉路は　２個と数えた。ｍ＝２０、ｎ＝１５　である。なお、分岐した枝道の先には必ず行き止まりがあるので、分岐点を数えることは行き止まりを数えることでもあり、ここで数えた分岐点の数は行き止まりの数より　１　多いだけである。

x　　回	１	２	３	４	５	６	７	８	９	１０	平均
３４	９０	８６	９３	９１	８０	８９	９０	８６	８５	８６	７９.８
５８	８６	９０	８９	７８	７８	８１	９１	８３	８７	８０	８４.３
８２	７９	７８	７７	７７	７７	８２	８０	７７	７７	８８	７９.２
１０４	７６	７１	７０	８２	７５	７６	７３	７０	７４	７６	７４.３
１２８	５９	６５	６６	６３	６４	６４	６９	６３	６２	６６	５７.８
１５２	５３	５８	５９	５５	５７	６０	４９	５９	５７	５７	５６.４

表１　解となる道の長さを変えて、分岐点の数を１０回ずつ数えた。

もし難易度が ｘ（ｍｎ－ｘ） で与えられるとすると、この値は ｘ＝ｍｎ／２ のとき最大となる。この値は今 ２０×１５／２＝１５０ なので、上の表によると、分岐点の数はかなり少なくなるだろう。従って、難易度が ｘ（ｍｎ－ｘ） で与えられるということはなさそうである。

分岐点の数は、ｘ が大きくなるとだんだん減少していって、最後には ０（ｍ、ｎ共に偶数のときは １）になることが明らかであるが、上の表で ｘ＝５８ のとき最大になっていることが注目される。ｍ＋ｎ－１＝３４ が解である道の長さの最小値であるが、解である道の長さがこれよりやや長いところが、調べなければならない分岐点・枝道が多くなり、最も難しくなると言えるのかもしれない。

ここで、またひとつの疑問が浮かび上がる。

［４］縦 ｍ個、横 ｎ個の格子点からなる迷路において、分岐点の数の最大値はいくらなのだろうか？

これも私には解けない。

５．出口

しかし、難易度についてさらに考えてみれば、もともと難しさというものはその解き方に依るであろう。例えば、常に左手を壁につけながら進むと、何も考えることなく必ず迷路から抜けることができるが、この場合、左側の壁の長さをできるだけ長くすることが、迷路から抜けるのに要する時間を長くすることになる。しかし、これを「難しい」と言えるだろうか？ 否である。結局、「難易度の調節」を目指したこと自体が誤りであったかもしれない。

６．帰り道

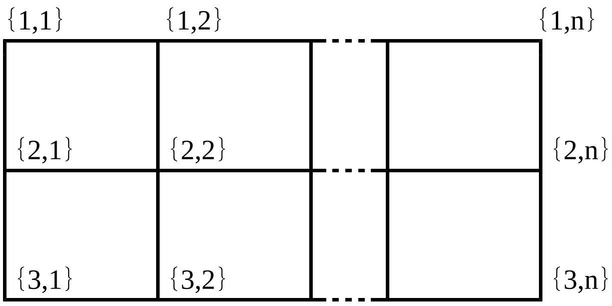
上記［１］の課題、すなわち、『縦 ｍ個、横 ｎ個の格子点からなる迷路において「最長の道」は何通りあるか？』はとうてい解けないと考えていたが、具体的な ｍ、ｎ についてコンピュータで数え上げることならできると思い付き、実際にそのためのプログラムを作成した。そして、ｍ、ｎ の少なくとも一方は奇数でなければならないので、ｍ＝３ に固定し、ｎ＝３，４，５ の場合について計算してみたところ、２，４，８ という結果を得た。ｎ＝２ のときは １ であるので、これも併せてこの結果を見ると、ｍ＝３ のときは 2^{n-2} 通りではないか、という予想もできそうではあるが、1999 年の本誌 7(2) に私が投稿した「円板の分割」で示

したように、１，２，４，８，１６，３１，・・・となる例もあるので、安易な予想はできない。そこで、もっと格子数を増やして計算したいのであるが、残念ながら、時間がかかり過ぎて、事実上できなかった。そこで、数え上げアルゴリズムについて相談しようと、南山大学情報理工学部教授　杉浦　洋氏（私と大学の同窓、私は物理学科、彼は数学科）に卒業以来の連絡を取った。すると、彼は、「３×ｎ格子については 2^{n-2} 通りと証明できるね。」と言って、その証明を書いて寄越した。その証明は複雑な漸化式を使う、やや理解しにくいものであったので、私が新しいアイディアを出して漸化式を簡単なものにした。すると、彼は元々のアイディアと私のアイディアを合わせて、さらに証明をすっきりしたものにした。これを以下に掲げる。

右の格子で、{1,1}から出発し、全ての交点を一度ずつもれなく通り、{3,n}に至る経路の総数を P_n とすると、

$$P_1=1、P_n=2^{n-2}\ (n \geqq 2)$$

である。／／



（証明）

{3,1}から出発し、全ての交点を一度ずつもれなく通り、{3,n}に至る経路の総数を Q_n とする。

まず、$P_1=1$、$Q_1=0$ である。以下、$n \geqq 2$ のときを考える。

(1) {1,1}→{1,2} と東進する経路数

このような経路は{3,1}を通過するために{2,2}→{2,1}→{3,1}→{3,2}か{3,2}→{3,1}→{2,1}→{2,2}を部分経路として含む。これを{2,2}→{3,2}、{3,2}→{2,2}とショートカットすれば、{1,2}以東における{1,2}から{3,n}への最長経路がもれなく得られる。ゆえに、{1,1}→{1,2}と東進して{3,n}に至る経路数は P_{n-1} である。

(2) {1,1}→{2,1} と南進する経路数

この経路は{2,1}→{3,1}→{3,2}と進まざるを得ないことがすぐ分かる。この後、{3,2}から{3,n}への最長経路数は Q_{n-1} であるから、{1,1}→{2,1}と南進して{3,n}に至る経路数は Q_{n-1} である。

以上より、

$$P_n=P_{n-1}+Q_{n-1} \qquad (n \geqq 2)$$

同様の論理で、

$$Q_n=P_{n-1}+Q_{n-1} \qquad (n \geqq 2)$$

この２式より、

$$P_n=Q_n \qquad (n \geqq 2)$$

ゆえに、

$$P_n=2P_{n-1} \qquad (n \geqq 3)$$

ここで、$P_2=1$ は明らかだから

$$P_n=2^{n-2} \qquad (n \geqq 2)$$

である。／／

シミュレーションに使った自作の迷路自動作成ソフト「迷作」は、私のホームページ http://www7a.biglobe.ne.jp/~EDEL-plus/　の「ダウンロード」ボタンからダウンロードできます。

８８Ｍ０５　　ふじのとしひろ
徳島県立城東高校　088-653-9111