第3回ウディタ学会

2020/04/29

サカモトトマト (SAKAMOTO TOMATE)

２部構成での発表

- 第１部：パーセントに気をつけろ
  - Coffee Break : イベントIDについて
  - Coffee Break : 基本システム用変数について
- 第２部：計算時間とNP完全入門

第1部：パーセントに気をつけろ

きっかけ

今まで遊んだフリーゲームを遊んで

確率の怖さを知らない人が多そう

と思ったから

例題

例題―ポテトのパラドクス

質量の99%が水分であるポテト

これを、水分が98%になるまで乾燥させる

ポテトの質量はどのくらい減ったか？

例題―ポテトのパラドクス

質量の99%が水分であるポテト

これを、水分が98%になるまで乾燥させる

ポテトの質量はどのくらい減ったか？

答え：半減

例題―ポテトのパラドクス【解説】

答え：半減

ポテトを 100g とすると、水分 99g, その他 1g

乾燥後、水分が 98 %になるということは、

「その他」が 2% になる

	水分	その他
乾燥前	99%	1%
乾燥後	98%	2%

例題―ポテトのパラドクス【解説】

答え：半減

ポテトを 100g とすると、水分 99g, その他 1g

乾燥後、水分が 98 %になるということは、

「その他」が 2% になるが、「その他」の質量 1g は変化しない

	水分	その他
乾燥前	99%	1%
乾燥後	98%	2%

	水分	その他
乾燥前	99g	1g
乾燥後	?? g	1g

例題―ポテトのパラドクス【解説】

答え：半減

ポテトを 100g とすると、水分 99g, その他 1g

乾燥後、水分が 98 %になるということは、

「その他」が 2% になるが、「その他」の質量 1g は変化しない

	水分	その他
乾燥前	99%	1%
乾燥後	98%	2%

	水分	その他	合計
乾燥前	99g	1g	100g
乾燥後	49 g	1g	50g

同じようなことが
命中・回避率でもいえる

ポテトのパラドクス―命中率編

命中率が99%であるウルファール

命中率が1%だけ下がる武器を装備して、
命中率が98%になってしまった

攻撃失敗率はどれだけ増えたか？

ポテトのパラドクス―命中率編

命中率が99%であるウルファール

命中率が1%だけ下がる武器を装備して、
命中率が98%になってしまった

攻撃失敗率はどれだけ増えたか？

答え：1%➔2% の2倍！

ポテトのパラドクス―回避率編

回避率が2%であるP･G･エディ

回避率が9%であるター　　に それぞれ

「回避率が2%上昇」する装備をつけさせたら？

ポテトのパラドクス―回避率編

回避率が2%であるP･G･エディ

回避率が9%であるター　　に それぞれ

「回避率が2%上昇」する装備をつけさせたら？

2% (1/50) ➔ 4% (1/25)

9% (1/11) ➔ 11% (1/9)

ポテトのパラドクス―ドロップ編

「ゴブリン」のアイテムドロップ率が12%だと
なかなかドロップしなかったので　14%にしてみた

12% (1/8) ➔ 14% (1/7)
あまり変わらない……

ポテトのパラドクス―ドロップ編

「ドラゴン」のアイテムドロップ率が2%だと
なかなかドロップしなかったので　4%にしてみた

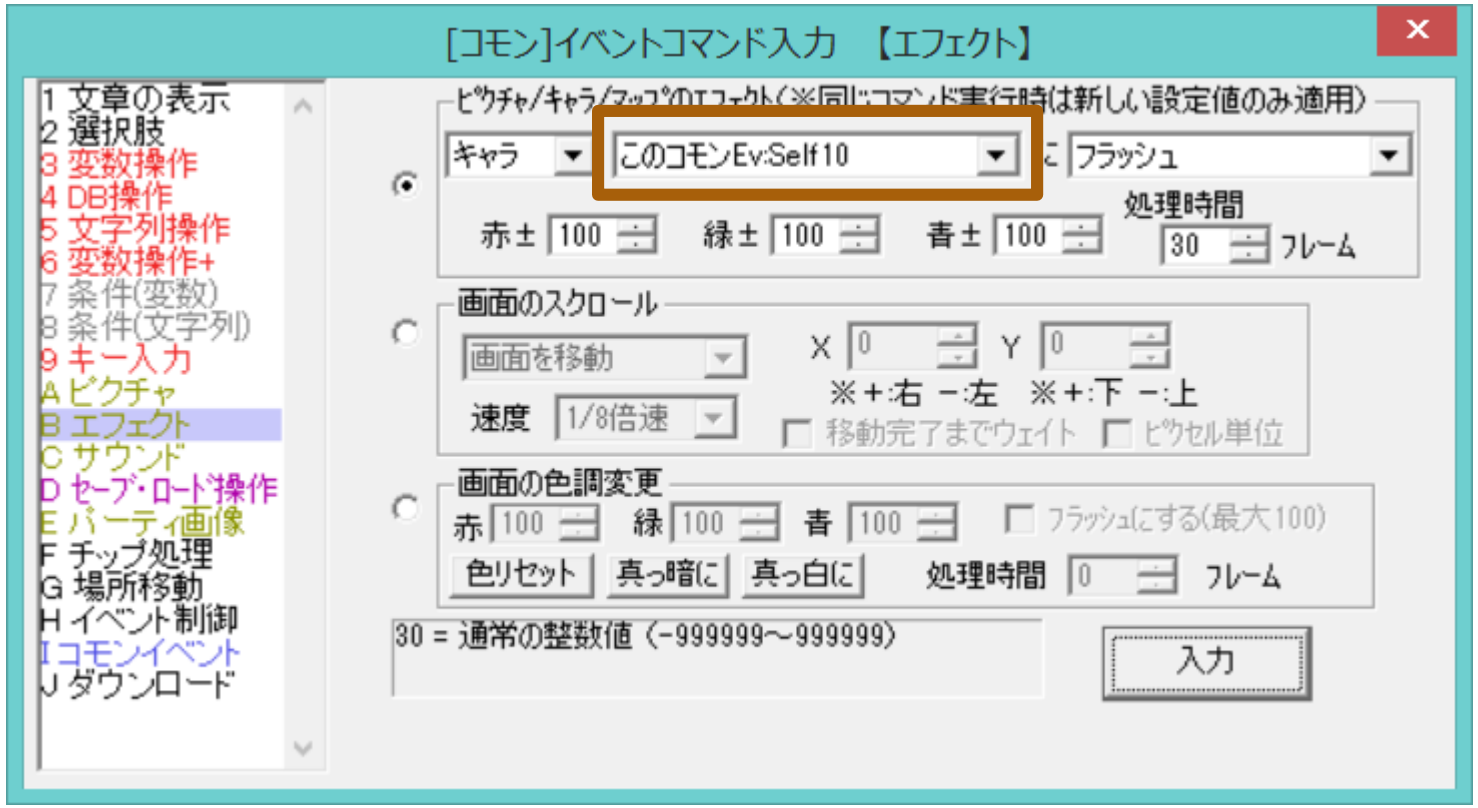
2% (1/50) ➔ 4% (1/25)
倍加！

第1部　まとめ

パーセントと上手に付き合って
バランスのいい戦闘レベルをデザインしよう！

Break：イベントIDについて

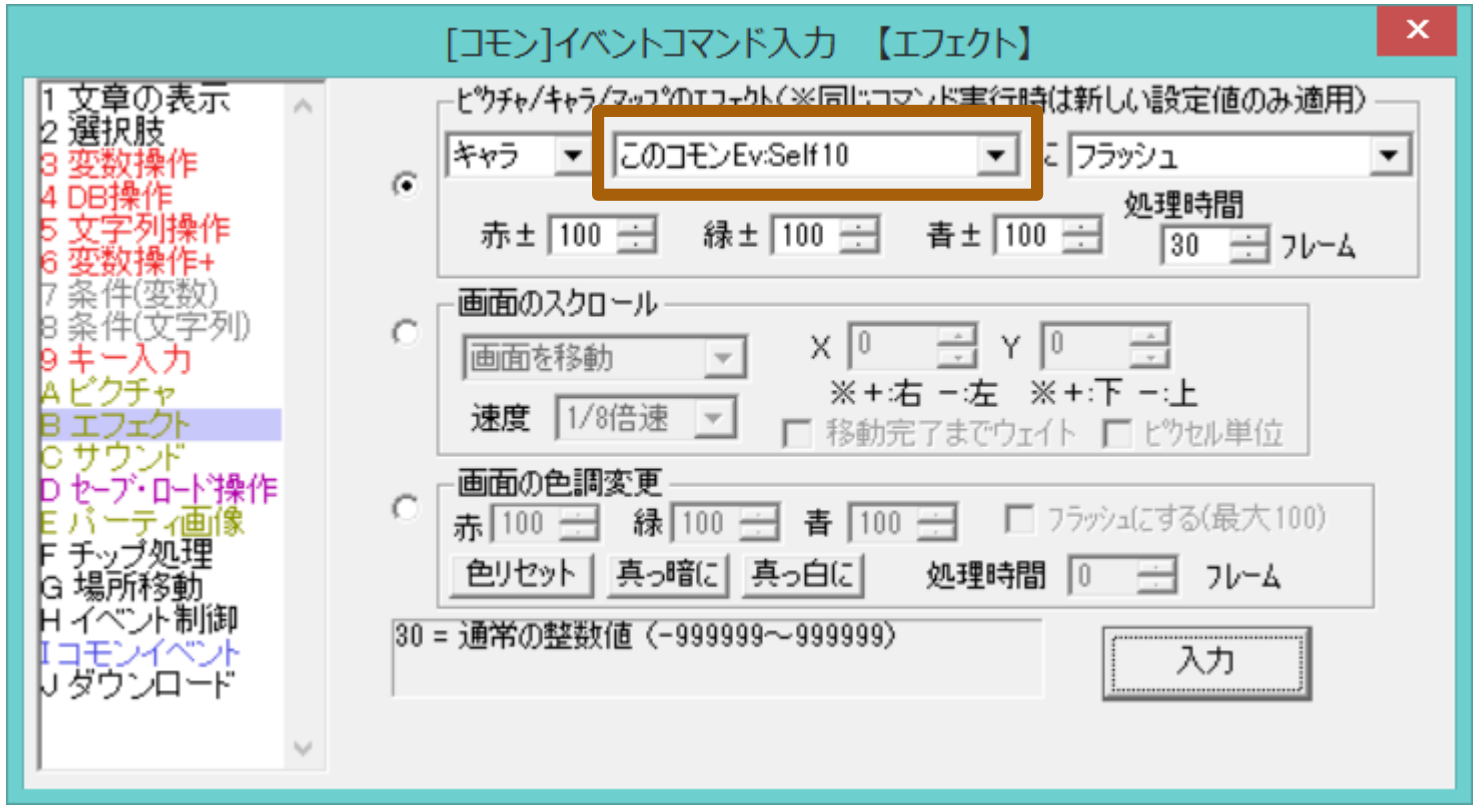
ID入力欄にマイナスを入れてみよう



この枠にイベントIDを入れることがあるかもしれませんが、
Cself10 という変数にしてみましょう。

Cself10 が
-1 ➔
-2 ➔
-3 ➔
-4 ➔
…

ID入力欄にマイナスを入れてみよう

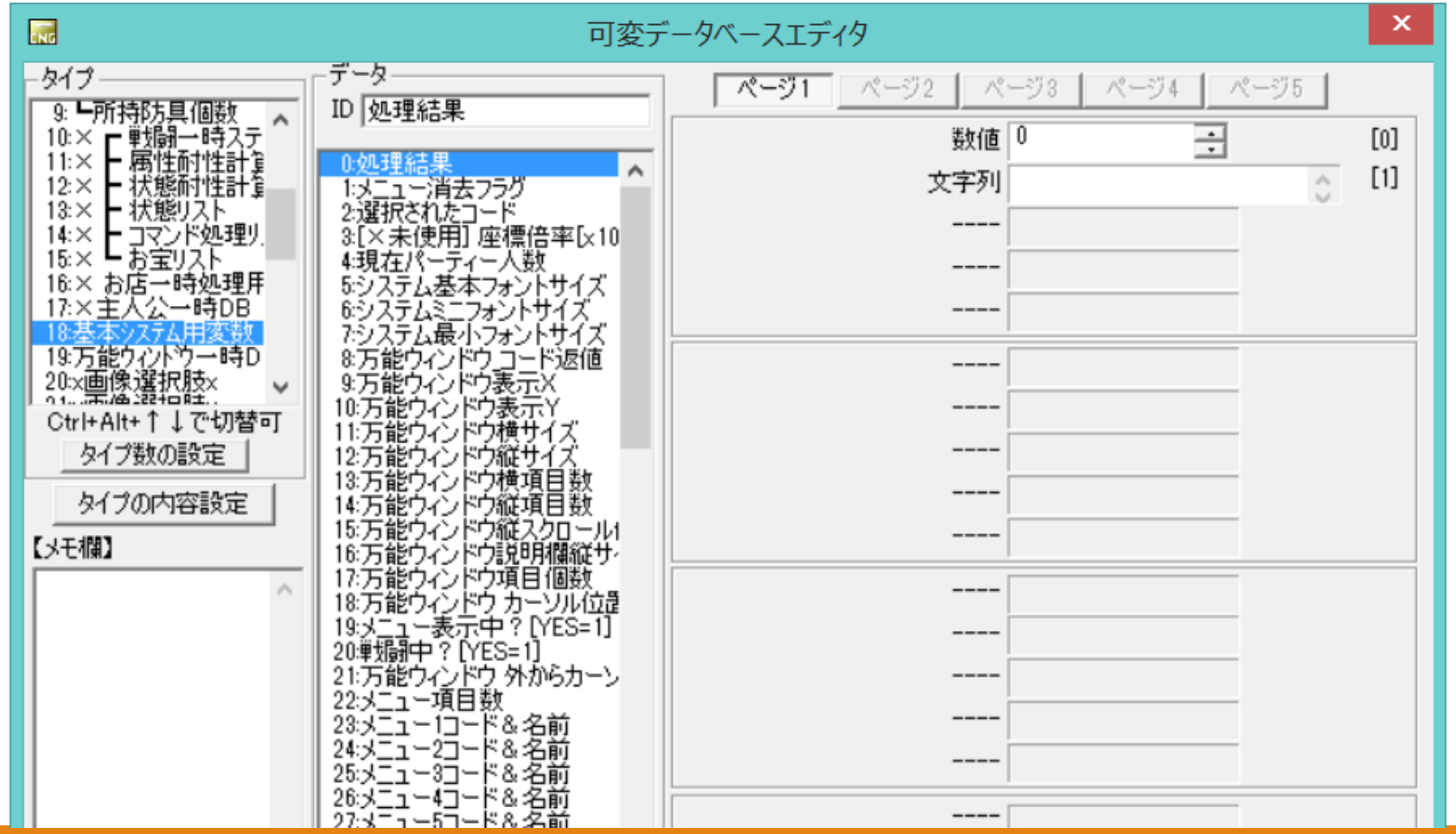


この枠にイベントIDを入れることがあるかもしれませんが、
Cself10 という変数にしてみましょう。

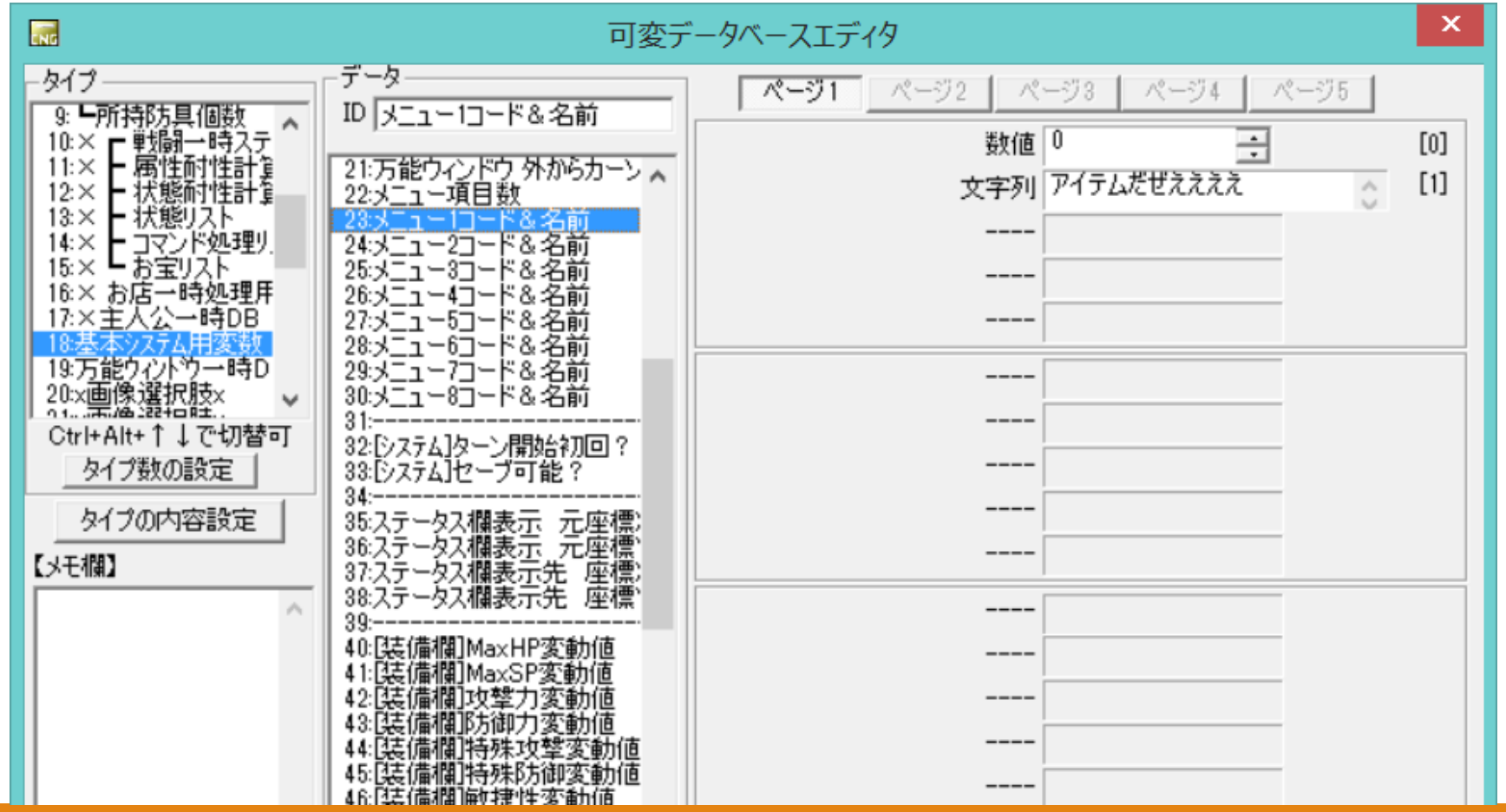
Cself10 が
-1 ➔ 呼び出し元のマップイベント
-2 ➔ 主人公
-3 ➔ 仲間1
-4 ➔ 仲間2
…

Break：基本システム用変数について

基本システムがもっと楽しくなる小ネタ

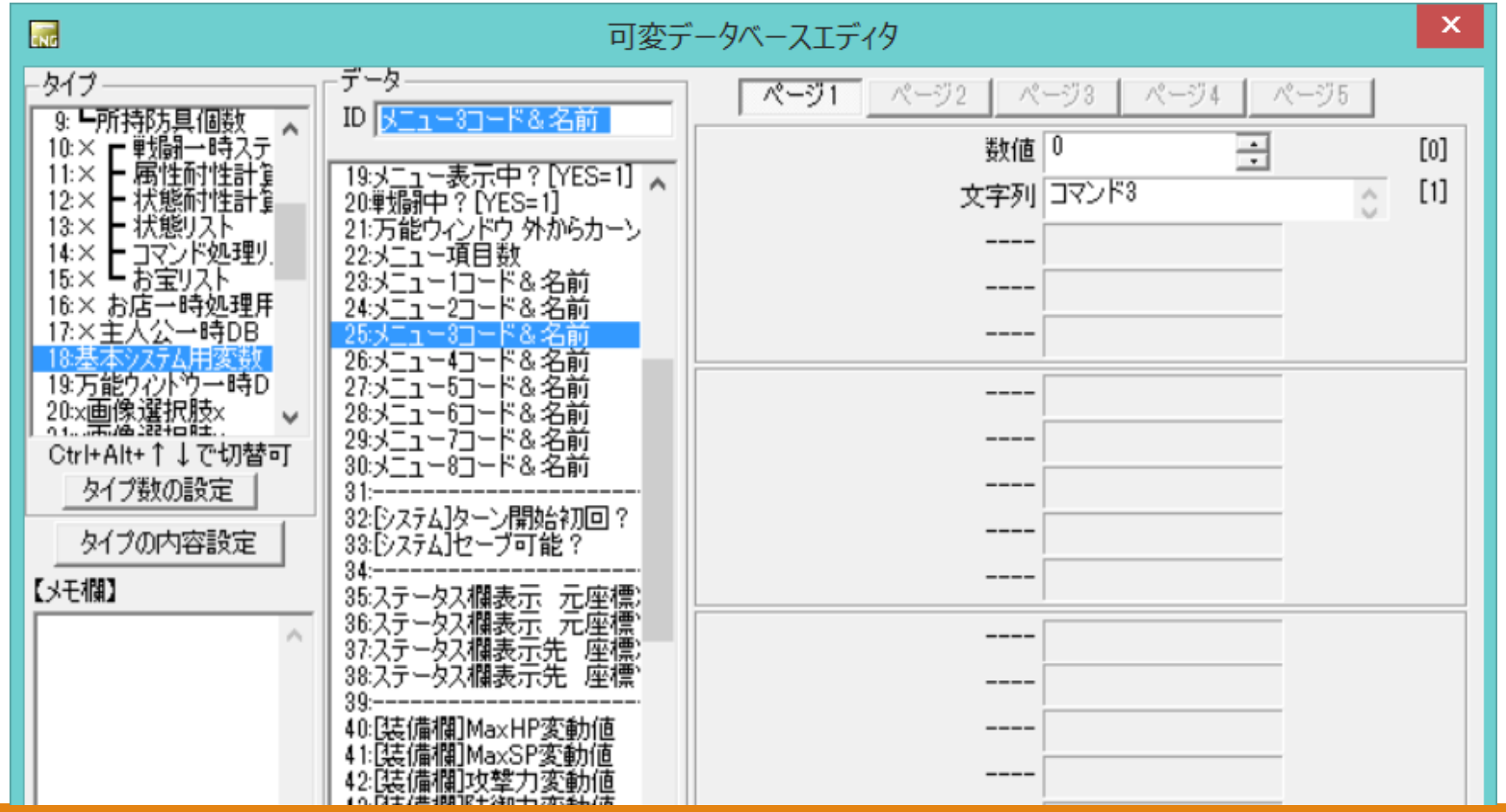


基本システムがもっと楽しくなる小ネタ



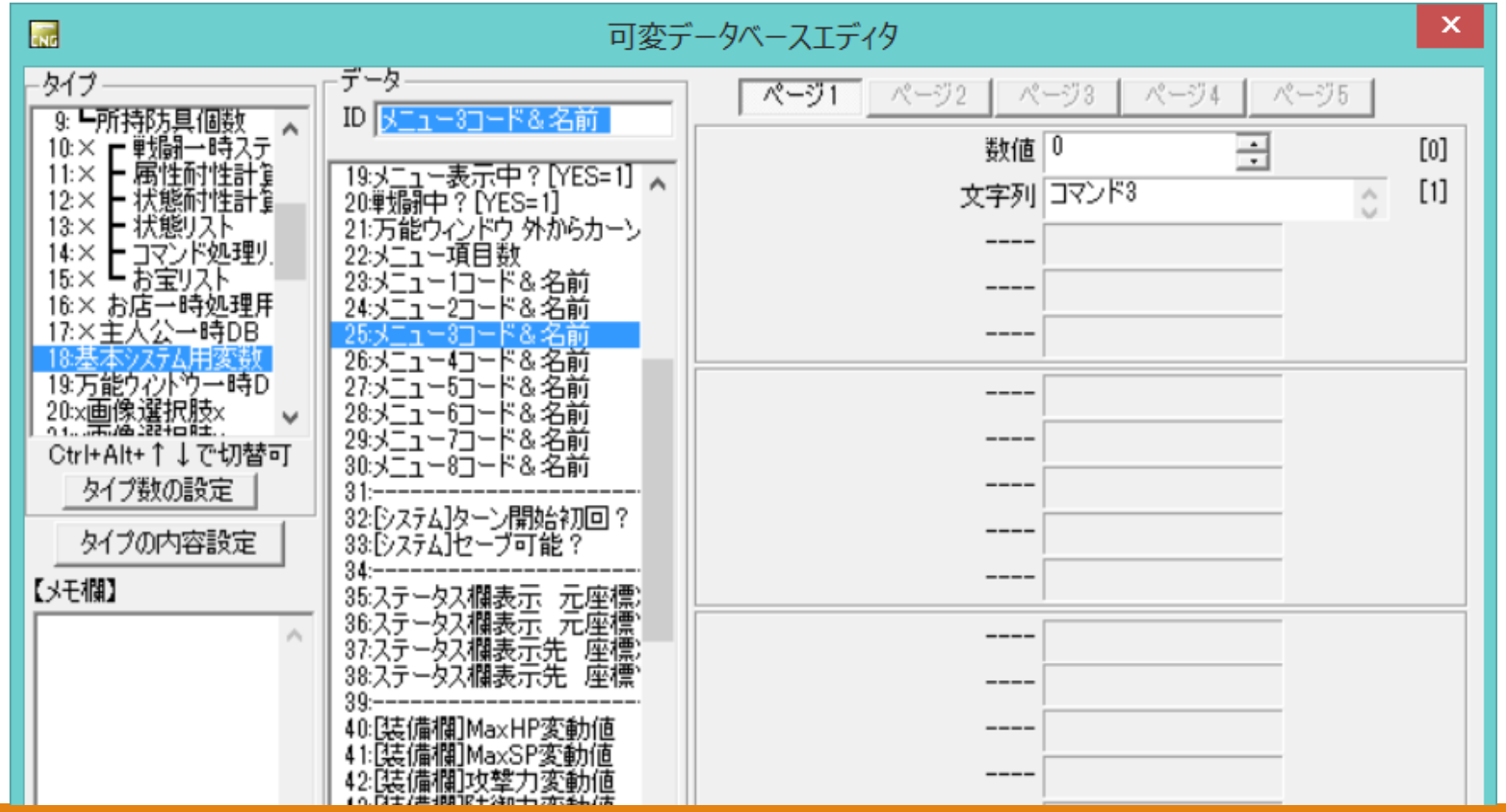
CDB[18:23:1]

基本システムがもっと楽しくなる小ネタ



CDB[18:25]

基本システムがもっと楽しくなる小ネタ



たとえば、CDBを操作するとき
「メニュー¥cself[10]コード＆名前」
という名前で呼び出すと
エラーが起こる場合があります

第2部：計算時間とNP完全入門

ウディタ　ほぼ　関係なし

きっかけ

システムエンジニアの方々と面談したけれど

プログラムの実行時間を考えるうえで重要なのに

「NP完全」をあまり知らない

人が多く、危機感を覚えたから

事実

NP完全に分類される問題を
「効率的に解く方法」は
未だ見つかっていない

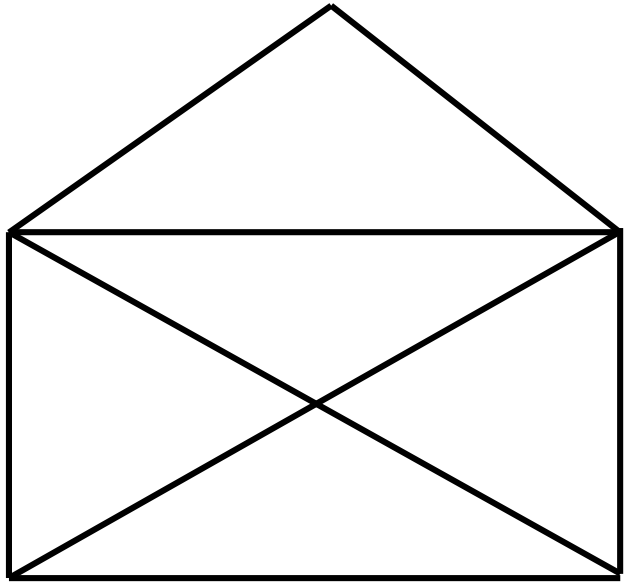
事実　つまり

コンピュータで計算させたい内容によっては、
サイズ（＝データ数）がでかいときに
時間内に解けるプログラム（＝コモン）を
作る方法が知られていない。
作れたら100万＄がもらえる。

例題

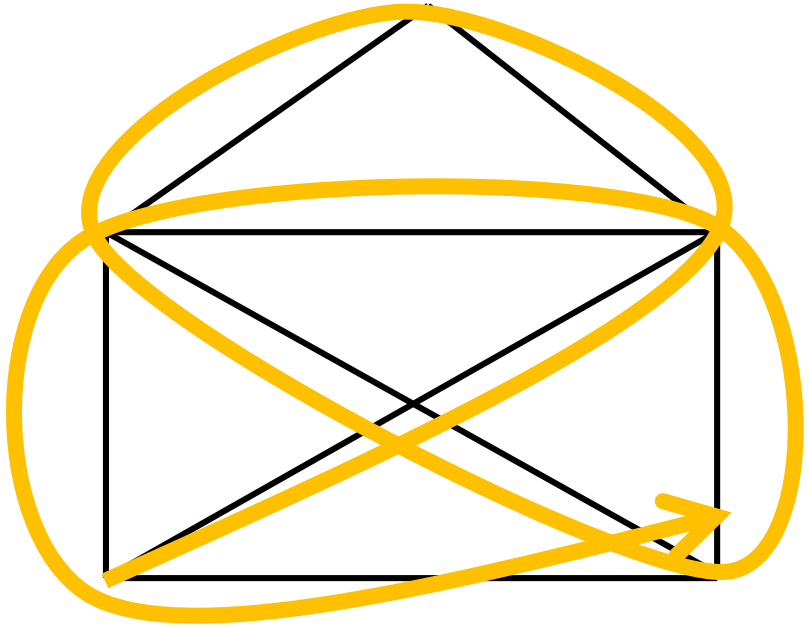
例題―一筆書き可能性判定問題1

この図形は一筆書きできるか？



例題―一筆書き可能性判定問題1

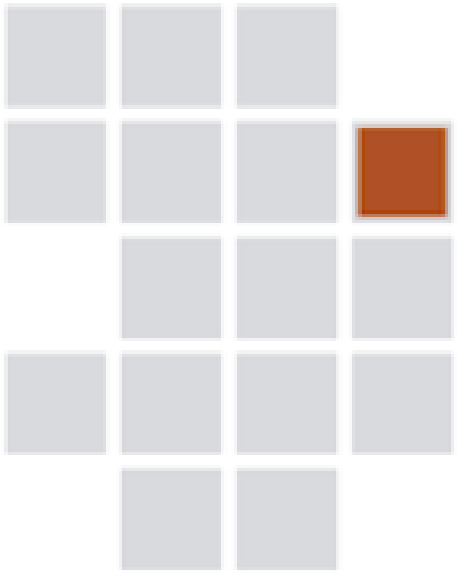
この図形は一筆書きできるか？　できる



例題―一筆書き可能性判定問題2

色マスからタテヨコに移動することで、

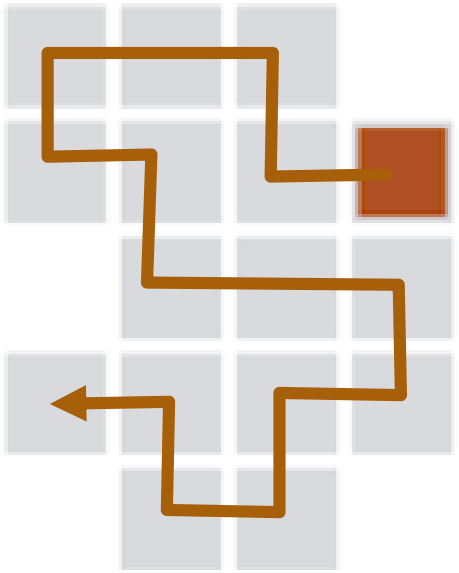
すべてのマスを1度ずつ通ることができるか？



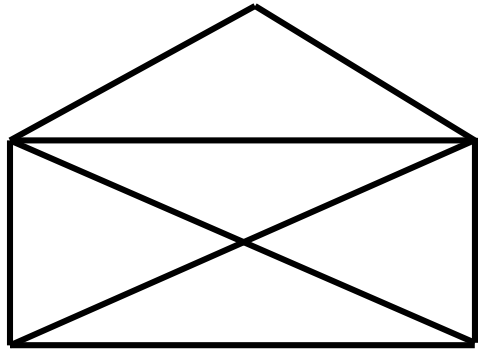
例題―一筆書き可能性判定問題2

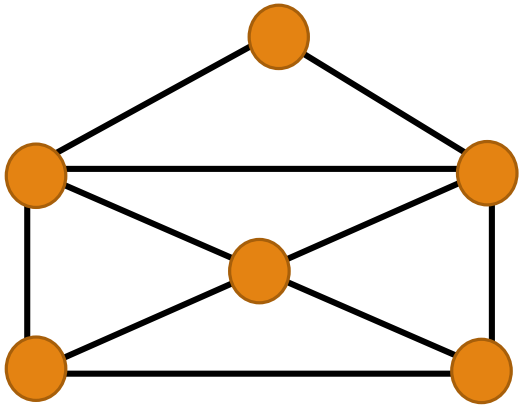
色マスからタテヨコに移動することで、

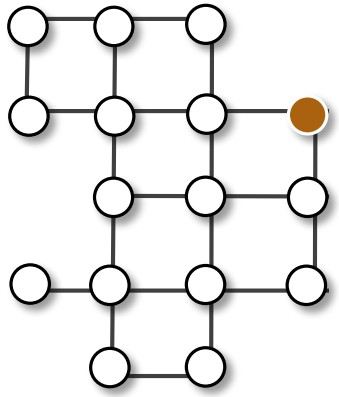
すべてのマスを1度ずつ通ることができるか？　できる



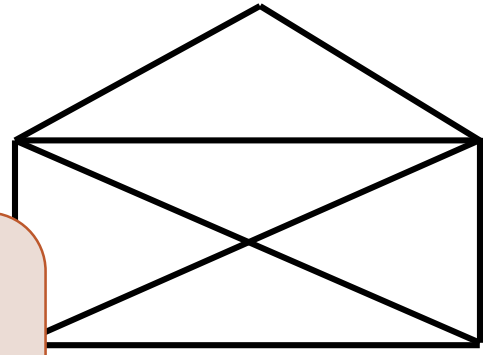
一筆書きには2種類存在する





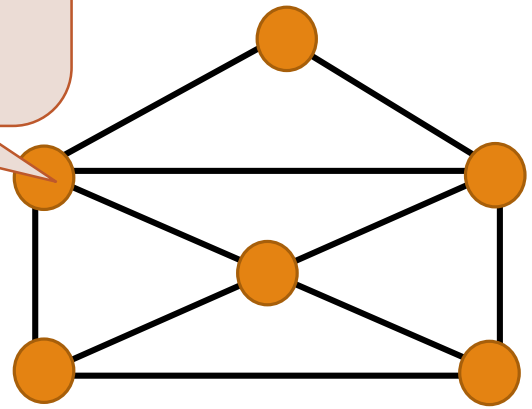


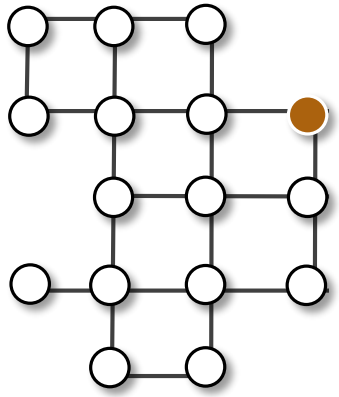
一筆書きには2種類存在する



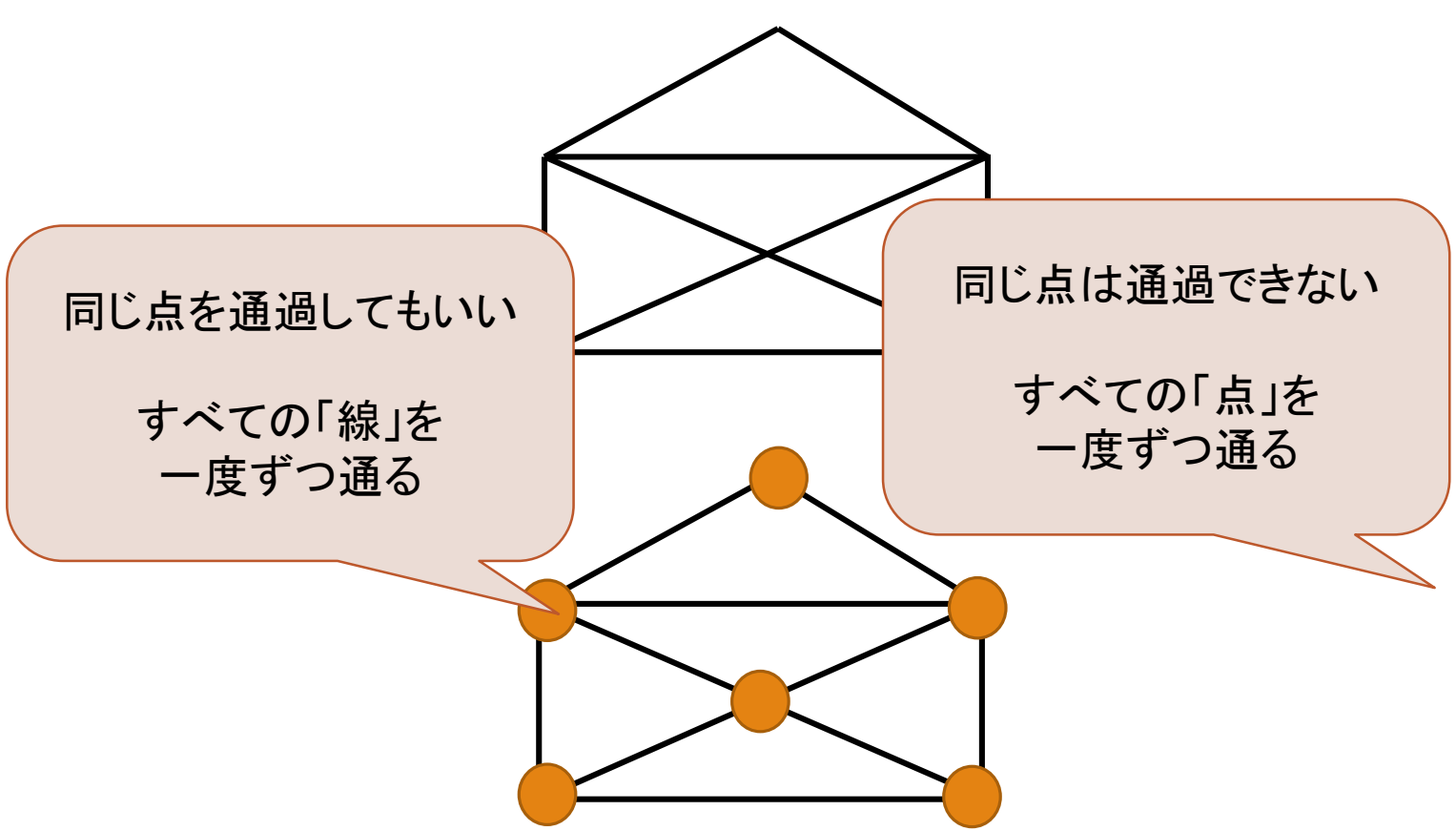
同じ点を通過してもいい

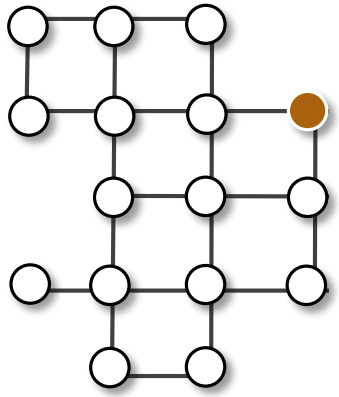
すべての「線」を
一度ずつ通る





一筆書きには2種類存在する

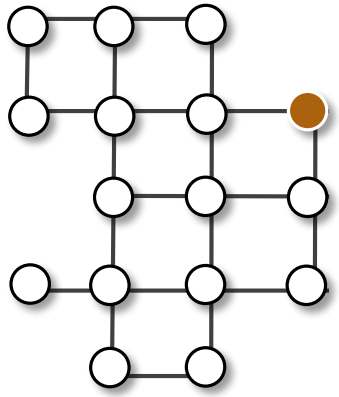




一筆書きには2種類存在する

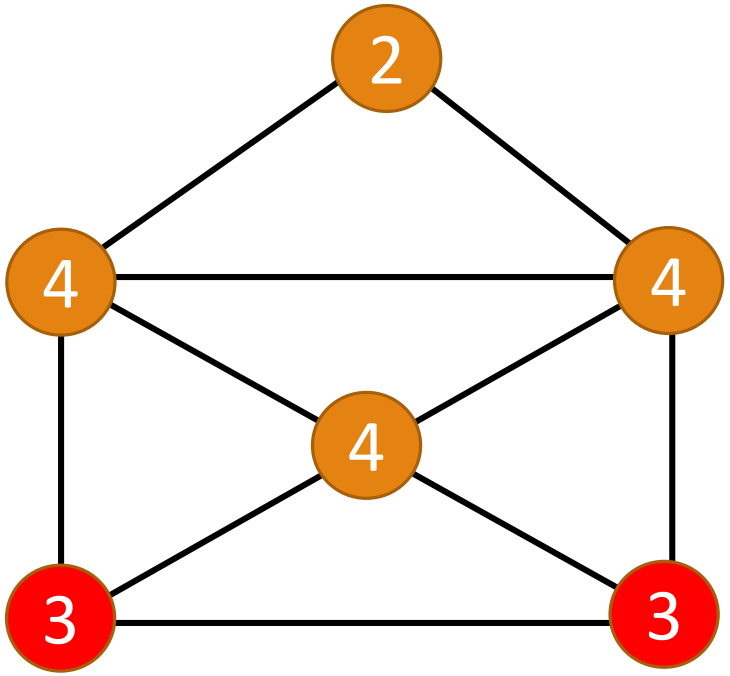
同じ点は通過できない

すべての「点」を
一度ずつ通る



線の一筆書き可能性

判定方法：交差点から出る線の数の偶奇を調べる

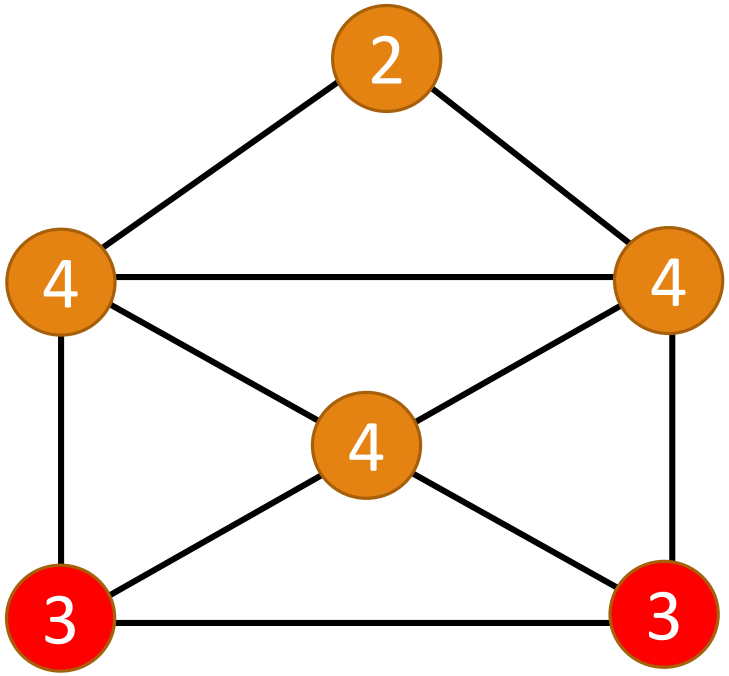


奇数の点が0個か2個だったら
一筆書きできる

そうでなければできない

線の一筆書き可能性

判定方法：交差点から出る線の数の偶奇を調べる



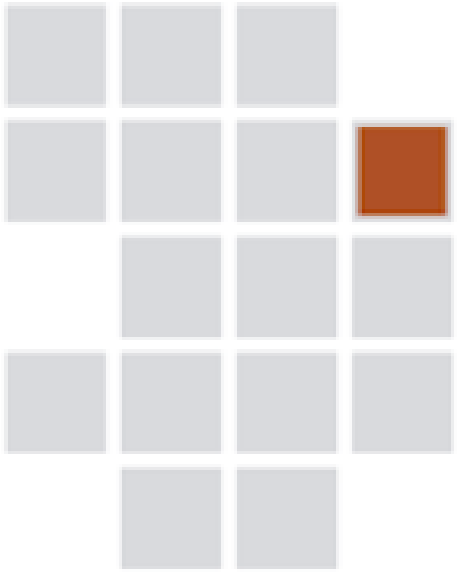
点の数を n とすると
n^2 に比例する行数の関数（コモン）で
確認できる

奇数の点が0個か2個だったら
一筆書きできる

そうでなければできない

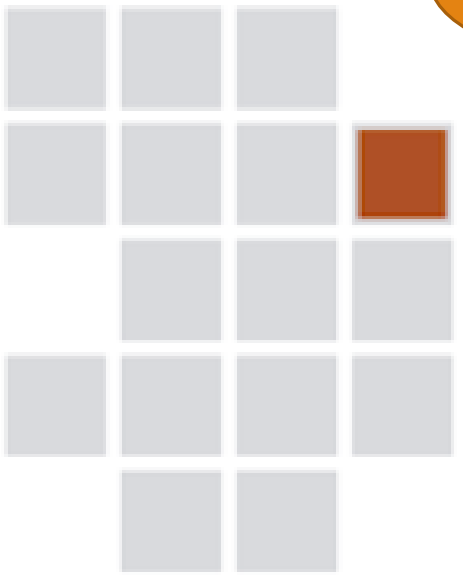
点の一筆書き可能性

判定方法：地道に探すしかない



点の一筆書き可能性

判定方法：地道に探すしかない



できる

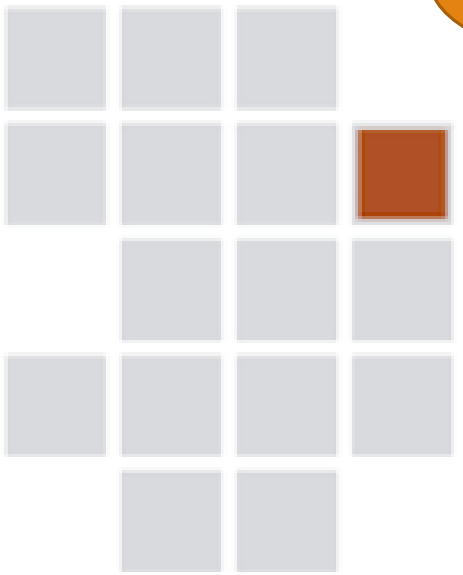
できない

この「できる」「できない」を
判定する「効率的な方法」は
現在まで知られていない

NP完全

点の一筆書き可能性

判定方法：地道に探すしかない



できる

できない

この「できる」「できない」を
判定する「効率的な方法」は
現在まで知られていない

現在の世界最小記録は
1.4^n に比例する行数の関数

なぜ「効率的」でないのか？

線の一筆書きは n^2 行程度で判定できる

点の一筆書きは 1.4^n 行程度で判定できる[NP完全]

なぜ「効率的」でないのか？

線の一筆書きは n^2 行程度で判定できる

点の一筆書きは 1.4^n 行程度で判定できる[NP完全]

n=50 とすると

50^2 =2500 だが　1.4^50 2≒20248916= 2025万

ウェイトなしだと 点の一筆書きでは40回「50万ループエラー」が発生することになる

なぜ「効率的」でないのか？

線の一筆書きは n^2 行程度で判定できる

点の一筆書きは 1.4^n 行程度で判定できる[NP完全]

n=50 とすると

50^2 =2500 だが　1.4^50 2≒20248916= 2025万

ウェイトなしだと 点の一筆書きでは40回「50万ループエラー」が

発生することになる

NP完全の問題を n^p (pは定数)で解く方法は見つかっていない

第2部：まとめ

世の中の問題の一部には「NP完全」という性質があり、
その性質をもつ問題を効率よく解く方法は見つかっていない。
つまり、より「NP完全」の問題を高速に計算するコモンを作ろうとしても
それが実装できる可能性は絶望的に低い。

「NP完全」に含まれる問題の例：
紹介した 点の一筆書き（ハミルトンパス問題）の他にも
テトリス全消し可能判定、ぷよぷよ成功可能判定、ペグソリティア成功可能判定、
トランプの各種ソリティア成功可能判定　など