プログラミング実習

第8回：簡単なアルゴリズム

清水 哲也（ shimizu@info.shonan-it.ac.jp ）

今回の授業内容

- 前回の課題の解答例
- 素数判定法
- ユークリッドの互除法
- 場合の数とアルゴリズム
- 課題

今回　内容　下　内容　粋

問題解決のための「アルゴリズム×数学」が 礎からしっかり身につく本

前回の課題の解答例

前回の課題の解答例

SA　学生　　　　　　　　解答例　　.

https://shimizu-lab.notion.site/2024-11826a533567807390dcfa0a5e288e15?pvs=4

素数判定法

素数判定法

自然数 N 素数 定 題

素数 定 下 介

- 単 素数 定
- 高速 素数 定

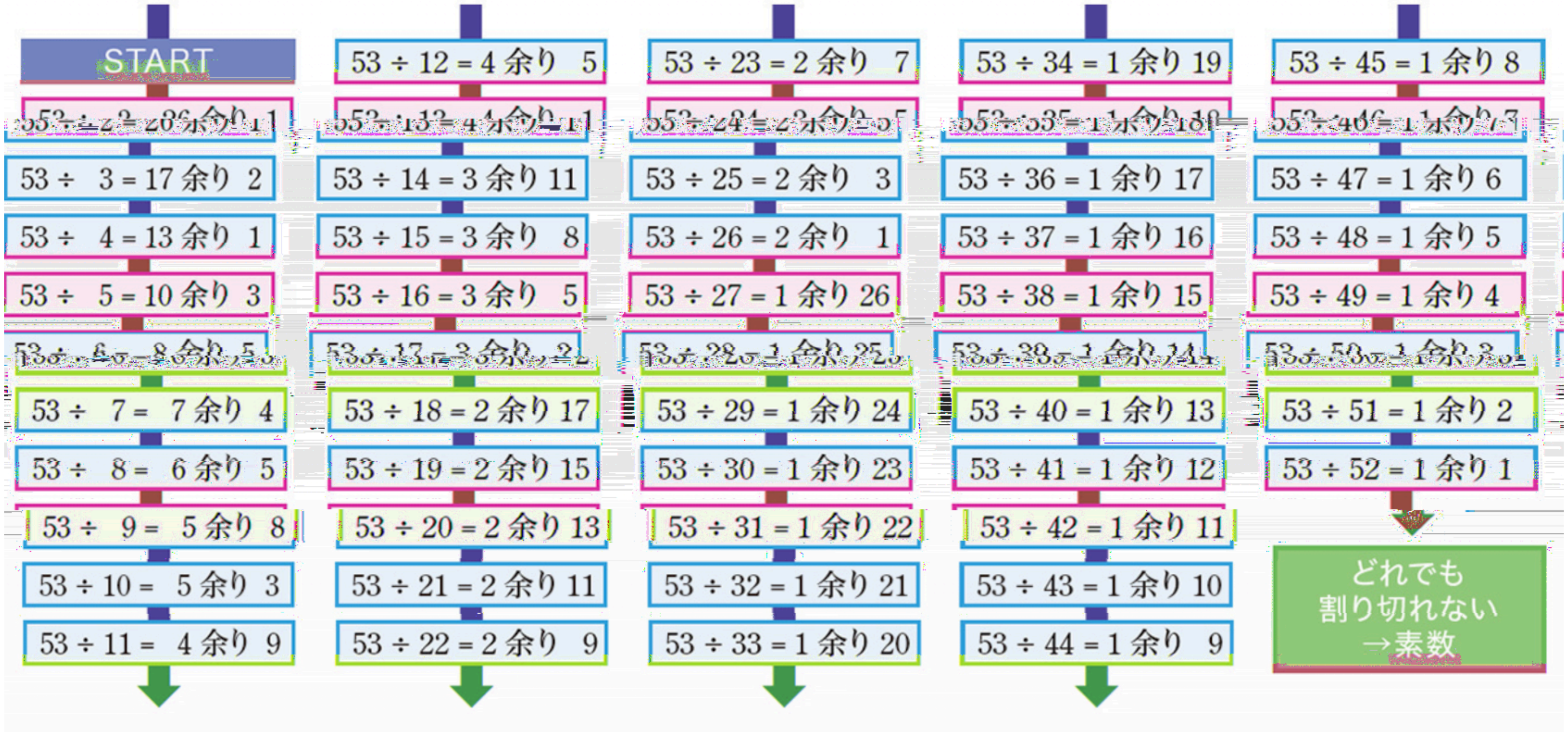
単純な素数判定法

例：53 素数 定

- 単 考 2 52 割 切 調
- 計算 時
- 整数 N 同 2 $N-1$ 割 切
  素数 定 行
- 計算量 $O(N)$ 時

単純な素数判定法

例：53　素数　　　　　　定



単純な素数判定法

```
static boolean isPrime(int N) {
        N を 0 の整数とし， が素数であれば が素数でなければ を返す
    for (int i =  ; i <= N -  ; i++) {
        if (N % i ==  )
        return false;
    }
    return tr e;
}
```

`int` 実行

`int` -2,147,483,648 ～ 2,147,483,647 ，例 「2147483647」 素数

定

数 実行

javaで扱える整数型

java 整数型 下 4 類

データ型	ビット数	値
`byte`	8 bit	-128 - 127 (3桁)
`short`	16 bit	-32768 - 32767 (5桁)
`int`	32 bit	-2147483648 - 2147483647 (10桁)
`long`	64 bit	-9223372036854775808 - 9223372036854775807 (19桁)

大 整数 一 型 `int` `long` 更 要

試 「9999999999989」 素数 定 後

注 ！！(私 約32)

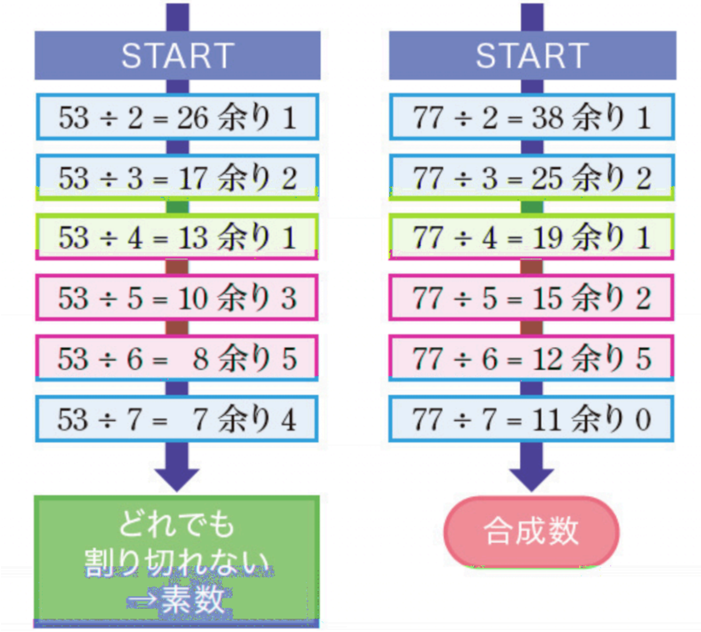
高速な素数判定法

単 素数 定 2 $N-1$ 全 調 N 大 計算
時
実 2 $N-1$ 全 調 要 2 $\sqrt{N}$ 調 割 切
N 素数 言
正 示 略

高速な素数判定法

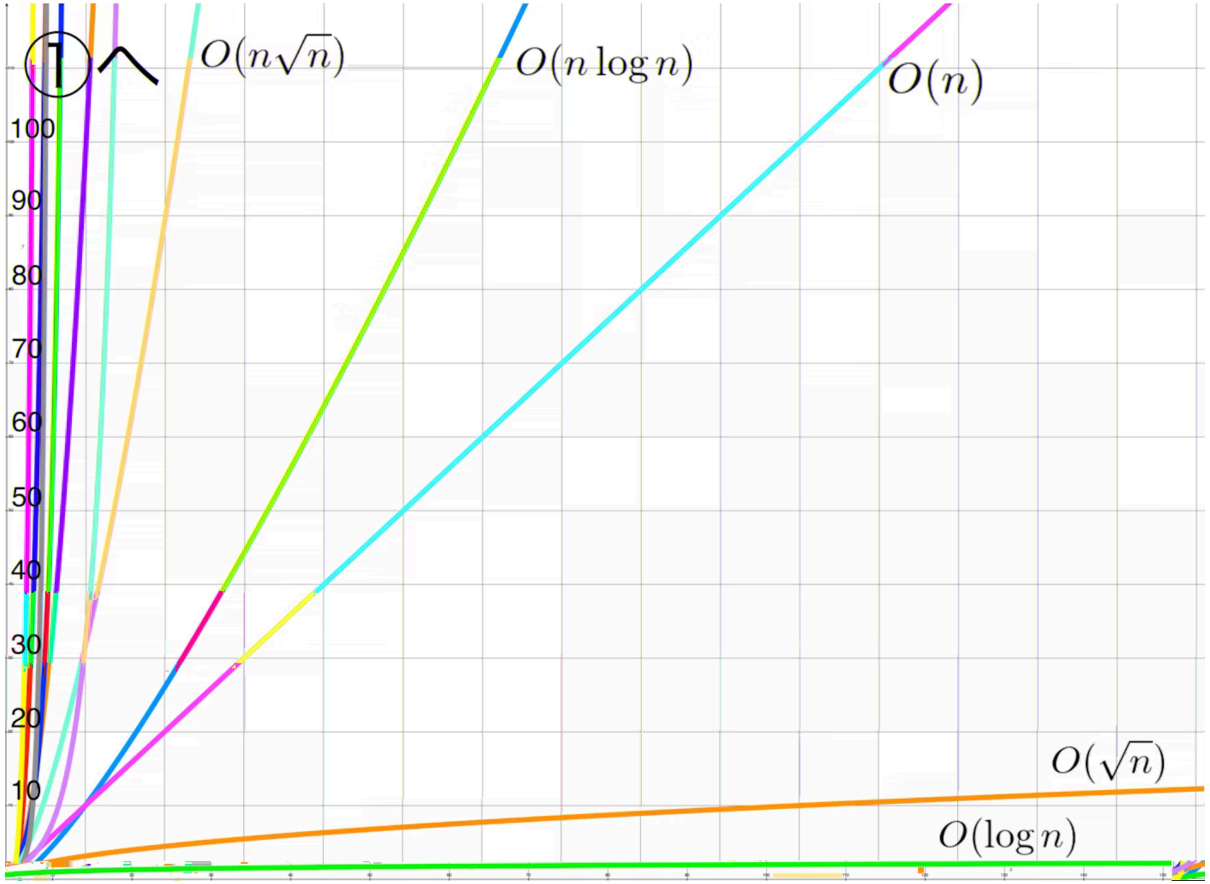
例　　53　　合 $\sqrt{53} = 7.28\cdots$　　$2, 3, 4, 5, 6, 7$　　割　切　　「53 は素数」

例　　77　　合 $\sqrt{77} = 8.77\cdots$　　$2, 3, 4, 5, 6, 7, 8$　　調　　, 7　割　切　「77は合成数」



高速な素数判定法

計算量　$O(\sqrt{N})$　，$O(N)$　較　示



https://qiita.com/kokorinosoba/items/1c7400ca6c740fe9f1ee

高速な素数判定法

```
static boolean isPrime(long N) {
      N を    0 の整数とし，  が素数であれば      が素数でなければ        を返す
    for( long i =  ; i * i <= N; i++) {
        if(N % i ==  ) return false;
    }
    return tr e;
}
```

，「999999999989」　素数　定　　計算時　　　　数ms程度

ユークリッドの互除法

ユークリッドの互除法

自然数 A 　 B 　最大公約数　　　　　題

素数　定　　同様　2　類　　　　　　　　実装

- 単
- 効率的　　　　　　　：　ー　　　　　互

ー　　　　　互　　　利用　　　　計算量　$O(\log(A+B))$

単純なアルゴリズム

33　88　最大公約数　計算

，答　33　下

件　利用　単　1　33　数　使　，33　88両　割

切　調

33 ÷ 1 = 33 余り 0	88 ÷ 1 = 88 余り 0	33 ÷ 13 = 2 余り 7	88 ÷ 13 = 6 余り 10	33 ÷ 25 = 1 余り 8	88 ÷ 25 = 3 余り 13
33 ÷ 2 = 16 余り 1	88 ÷ 2 = 44 余り 0	33 ÷ 14 = 2 余り 5	88 ÷ 14 = 6 余り 4	33 ÷ 26 = 1 余り 7	88 ÷ 26 = 3 余り 10
33 ÷ 3 = 11 余り 0	88 ÷ 3 = 29 余り 1	33 ÷ 15 = 2 余り 3	88 ÷ 15 = 5 余り 13	33 ÷ 27 = 1 余り 6	88 ÷ 27 = 3 余り 7
33 ÷ 4 = 8 余り 1	88 ÷ 4 = 22 余り 0	33 ÷ 16 = 2 余り 1	88 ÷ 16 = 5 余り 8	33 ÷ 28 = 1 余り 5	88 ÷ 28 = 3 余り 4
33 ÷ 5 = 6 余り 3	88 ÷ 5 = 17 余り 3	33 ÷ 17 = 1 余り 16	88 ÷ 17 = 5 余り 3	33 ÷ 29 = 1 余り 4	88 ÷ 29 = 3 余り 1
33 ÷ 6 = 5 余り 3	88 ÷ 6 = 14 余り 4	33 ÷ 18 = 1 余り 15	88 ÷ 18 = 4 余り 16	33 ÷ 30 = 1 余り 3	88 ÷ 30 = 2 余り 28
33 ÷ 7 = 4 余り 5	88 ÷ 7 = 12 余り 4	33 ÷ 19 = 1 余り 14	88 ÷ 19 = 4 余り 12	33 ÷ 31 = 1 余り 2	88 ÷ 31 = 2 余り 26
33 ÷ 8 = 4 余り 1	88 ÷ 8 = 11 余り 0	33 ÷ 20 = 1 余り 13	88 ÷ 20 = 4 余り 8	33 ÷ 32 = 1 余り 1	88 ÷ 32 = 2 余り 24
33 ÷ 9 = 3 余り 6	88 ÷ 9 = 9 余り 7	33 ÷ 21 = 1 余り 12	88 ÷ 21 = 4 余り 4	33 ÷ 33 = 1 余り 0	88 ÷ 33 = 2 余り 22
33 ÷ 10 = 3 余り 3	88 ÷ 10 = 8 余り 8	33 ÷ 22 = 1 余り 11	88 ÷ 22 = 4 余り 0	両方割り切れた最大の数は11 →最大公約数は11	
33 ÷ 11 = 3 余り 0	88 ÷ 11 = 8 余り 0	33 ÷ 23 = 1 余り 10	88 ÷ 23 = 3 余り 19		
33 ÷ 12 = 2 余り 9	88 ÷ 12 = 7 余り 4	33 ÷ 24 = 1 余り 9	88 ÷ 24 = 3 余り 16		

単純なアルゴリズム

```
    正の整数☒と☒の☒最大公約数を返すメソッド
      は☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒(☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒最大公約数）の略
static long GCD(long A, long B) {
  long Answer =  ;

  for (long i =  ; i <= Math.min(A, B); i++) {
    if (A % i ==   && B % i ==  ) {
      Answer = i;
    }
  }
  return Answer;
}
```

`Math.min(A,B)` A B 値

余 計算 $2 \times \min(A, B)$ 回行 要 , 効率的

100000000000 123450000000 試

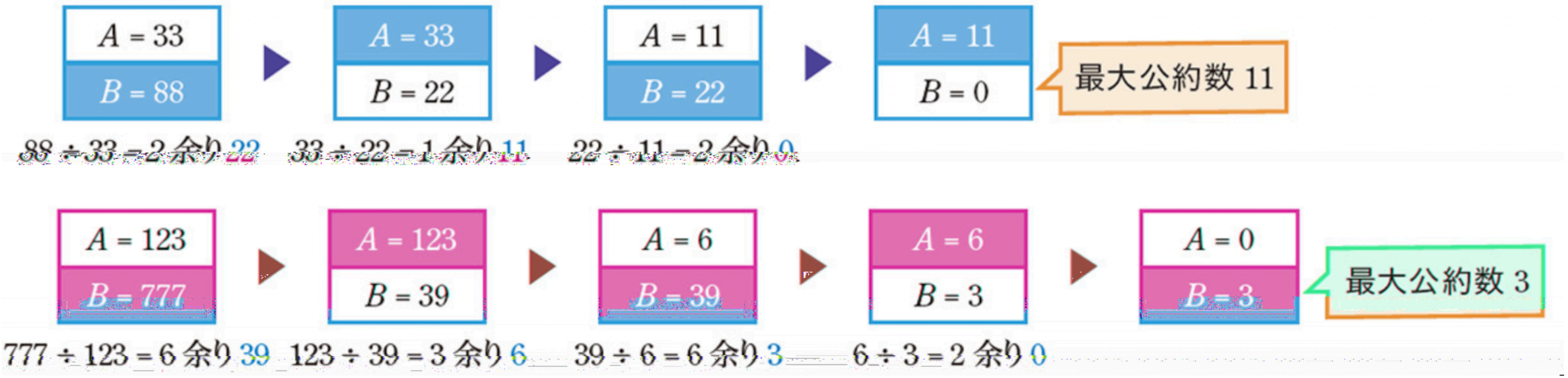
効率的なアルゴリズム：ユークリッドの互除法

下　　　　利用　　　2　　正整数　最大公約数　高速　計算

> 1. 大きいほうの数を「大きいほうを小さいほうで割った余り」に書き換えるという操作を繰り返す
> 2. 片方が0になったら操作を終了する．もう片方の数が最大公約数である

効率的なアルゴリズム：ユークリッドの互除法

33　88　最大公約数, 123　777　最大公約数　　計算　　下



ユークリッドの互除法

A　B　最大公約数　　計算量　$O(\log(A+B))$　　, A, B
10^{18} 程度　　程度　　計算

効率的なアルゴリズム：ユークリッドの互除法

```
static long EuclideanGCD(long A, long B) {
  while (A >=   && B >=  ) {
    if (A < B) {
      B = B % A;     A < B のÝ 合，大きい方⊠⊠ き換える
    } else {
      A = A % B;     A >= BのÝ 合，大きい方⊠⊠ き換える
    }
  }
  if (A >=  ) {
    return A;
  }
  return B;
}
```

A B 大 係 行 操作 , `if` 用

合 行

更 効率的 実装

場合の数とアルゴリズム

場合の数とアルゴリズム

，階　・二項係数・積　則　，　的　合　数　公式

- 公式1：積　則
- 公式2：積　則　拡張
- 公式3：n個　替　数　$n!$
- 公式4：n 個　r 個　${}_n\mathrm{P}_r$
- 公式5：n 個　r 個　選　${}_n\mathrm{C}_r$
- 応用例1：　数
- 応用例2：同　ー　組　合
- 応用例3：全探索　計算回数

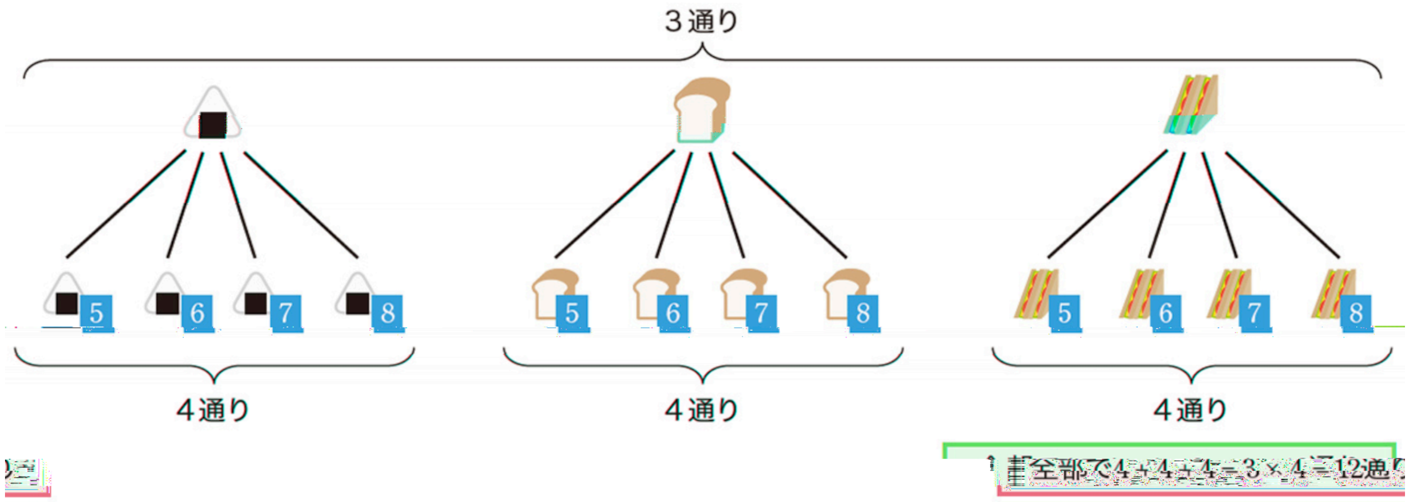
基本公式 1 ：積の法則

事　A　N 通，事　B　M 通，事　A, B

組　合　全　$NM(N \times M)$ 通

- 日　朝　，　・　・
- 日　時　，5:00, 6:00, 7:00, 8:00

朝　事　A,　時　事　B　$3 \times 4 = 12$ 通　，　積の法則

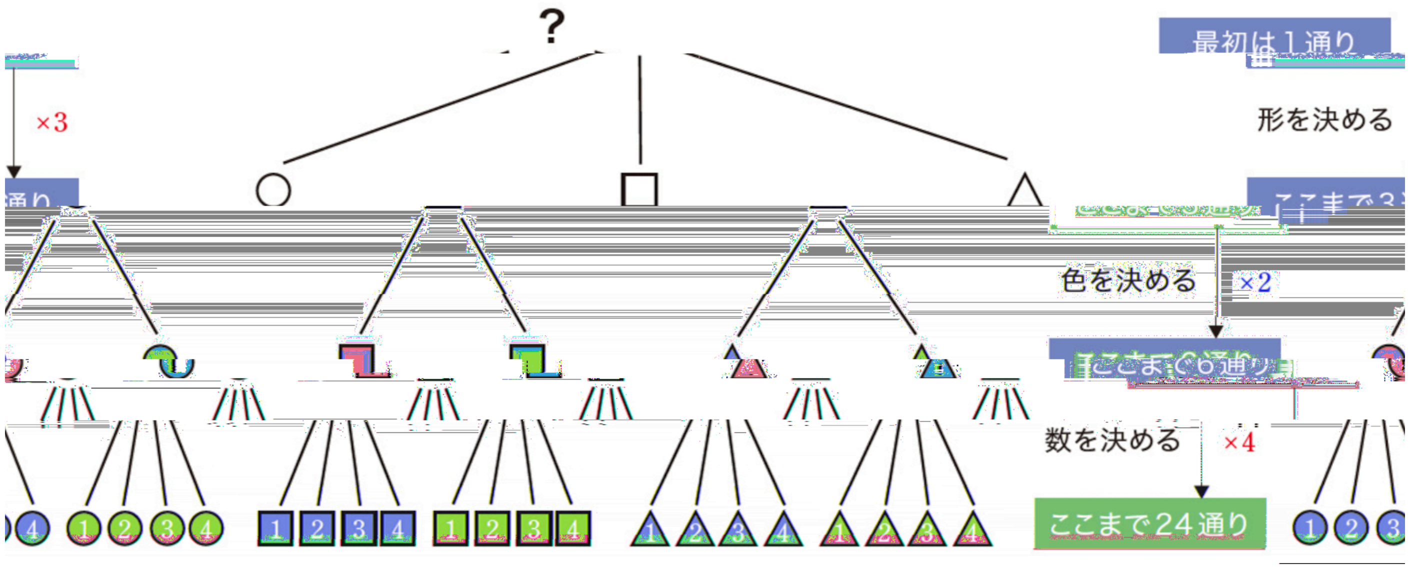


基本公式❷：積の法則の拡張

積の法則　事　３　合　拡張

- 形：円形，四角形，三角形
- 色：赤・青
- 記入する数字：1,2,3,4

合　組　合　数　積　則　$3 \times 2 \times 4 = 16$通



基本公式 2 ：積の法則の拡張

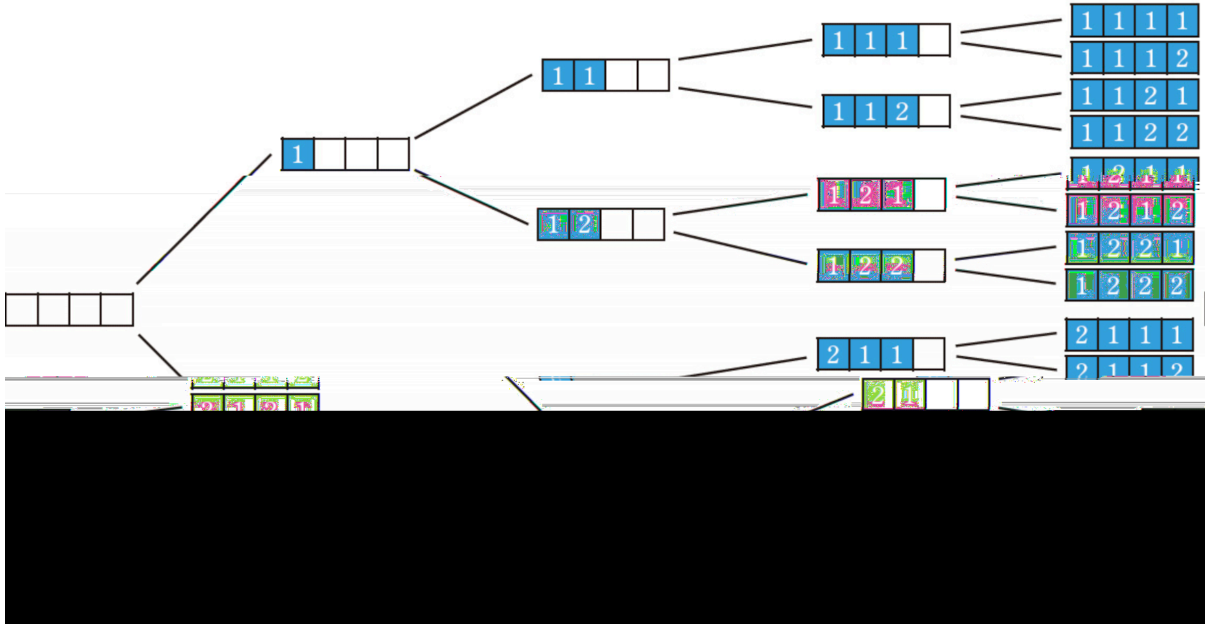
選択肢 同 数（例 M 通 ） 事 N 個 組 合 数

$$M \times M \times M \times \cdots \times M = M^N$$

通

例 , 要素 1 2 長 4 数列 $A = (A_1, A_2, A_3, A_4)$ 個数

$2^4 = 24$通



基本公式3： 個のモノを並び替える方法の数は

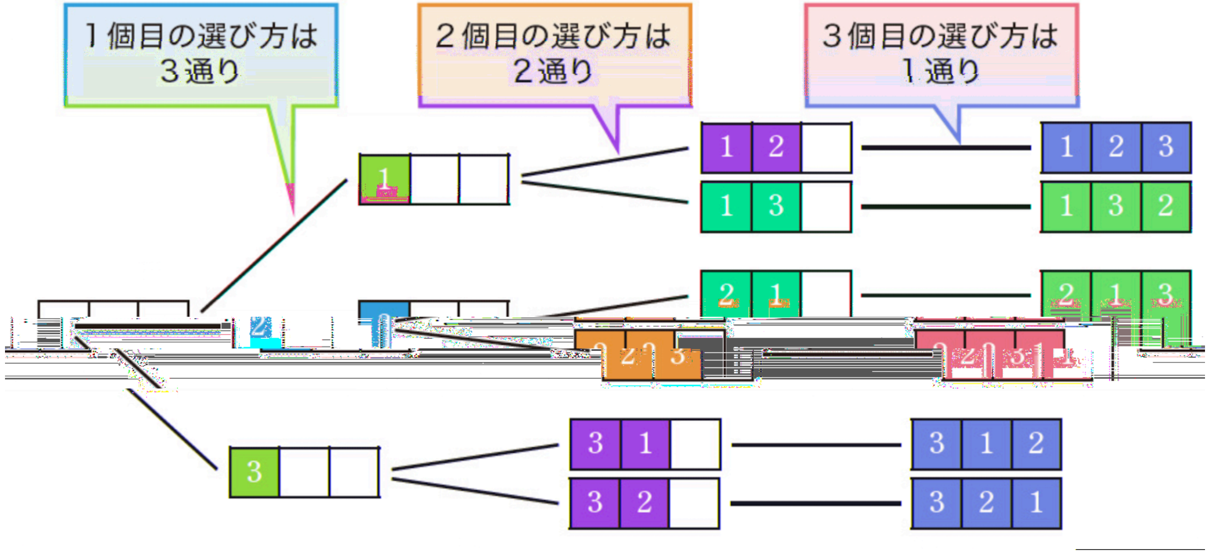
n 個　　　　　　　　替

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

通

例　　，３　　整数$1, 2, 3$　　　替　　　　　$3! = 3 \times 2 \times 1 = 6$通

$3!$ 通　　　　理由　　下　　形図



基本公式4：　個のモノから　個を並べる方法は

n 個　r 個　選，　列　数　下　式

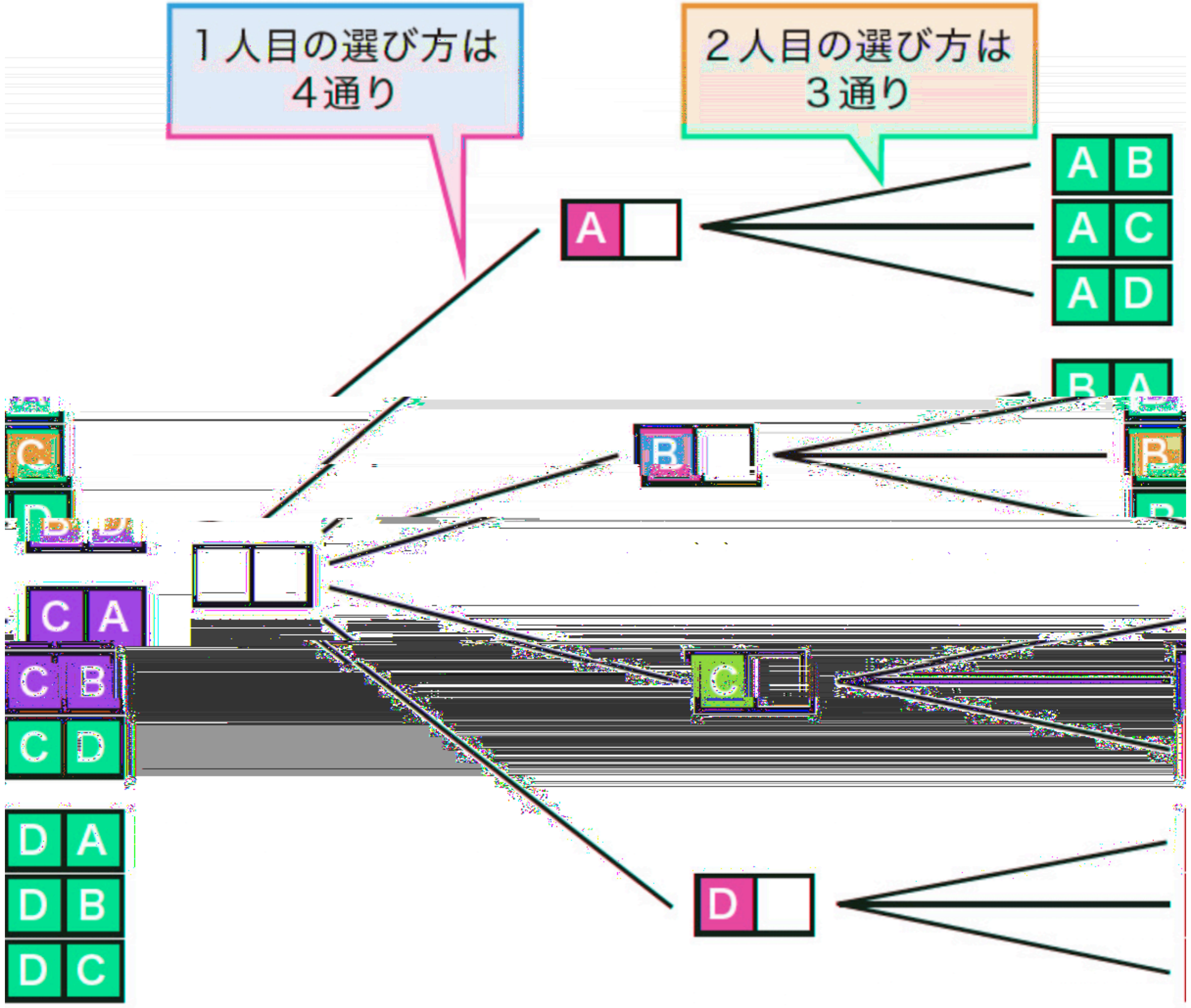
$$_n\mathrm{P}_r = \frac{n!}{(n-r)!} = n \times (n-1) \times (n-2) \times \cdots \times (n-r+1)$$

，人 A, B, C, D　中　2人　選，　決　$_4\mathrm{P}_2 = 12$ 通

「1人　選　n 通」「2人　選　$n-1$ 通」　考　積の法則　適用

式　導

基本公式4：　個のモノから　個を並べる方法は



基本公式5：

基本公式5：　個のモノから　個を選ぶ方法は

応用例1：買い物の方法の数

下　題　考

> コンビニには [illegible] 個の品物が売られており，[illegible] 番目（[illegible]）の商品の値段は [illegible] 円です．異なる2つの品物を買う方法のうち，合計値段が500円となるものは何通りありますか．
> 制約：[illegible] は [illegible] のいずれか

選　全探索　思　思．N 個　中　2 個　選，全 ${}_N\mathrm{C}_2$ 通　選．計算量 $O(N^2)$，効率．

応用例1：買い物の方法の数

全探索 考

合計値段 500円 仕 考 ， 下 2

方法A：100円 1個, 400円 1個

方法B：200円 1個, 300円 1個

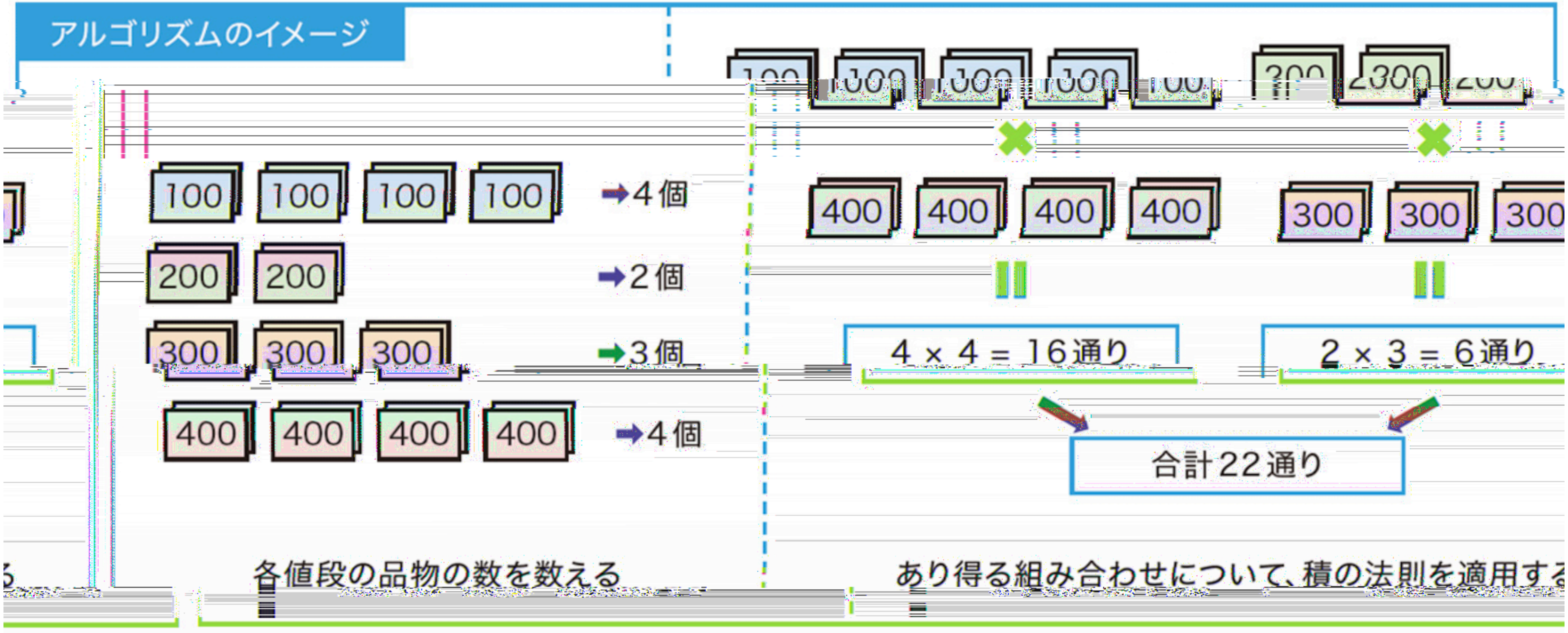
，100円・200円・300円・400円 数 a, b, c, d 個 ，積則 ，各 数 次 通 .

方法A：a 通 $\times d$通 = ad 通

方法B：b 通 $\times c$通 = bc 通

， 答 $ad + bc$. a, b, c, d 値 計算量 $O(N)$ 数 ，$N = 200000$ ー 1 内 答

応用例 1 ：買い物の方法の数



応用例2：同色カードの組み合わせ

一　　選　　個数　　　　題

> 　枚のカードがあり，左から　番目（　　　　　　）のカードの色は　　です．
> 　のとき赤色，　　　　　のとき黄色，　　　　　のとき青色です．同じ色のカードを2枚選ぶ方法は何通りありますか．
> 制約：　　　　　　　　　，

一　　選　　全探索　　　　思　　　　　　．　　2　　一　　選　　，全探索　　　　　計算量　$O(N^2)$　　　　　，効率

応用例 2 ：同色カードの組み合わせ

全探索 考

赤 ・黄 ・青 ー 数 x, y, z ， 下

.

- 赤 ー 2 選 ${}_x\mathrm{C}_2$ 通
- 黄 ー 2 選 ${}_y\mathrm{C}_2$ 通
- 青 ー 2 選 ${}_z\mathrm{C}_2$ 通

， 答 下 通 .

$$ {}_x\mathrm{C}_2 + {}_y\mathrm{C}_2 + {}_z\mathrm{C}_2 = \frac{x(x-1)}{2} + \frac{y(y-1)}{2} + \frac{z(z-1)}{2} $$

考 ，赤 ・黄 ・青 ー 数 数 答

. 計算量 $O(N)$ ，全探索 大 効率 良

応用例2：同色カードの組み合わせ

応用例3：全探索の計算回数

5 ー 選 題

> 枚のカードがあり，左から 番目（ ）のカードには整数 が書かれています．カードを5枚選ぶ方法のうち，選んだカードに書かれた整数の和がちょうど となるものは何通りありますか．
>
> 制約： ，

題 全探索 考 ．今回 5 ー 選 ，全探索 計算量 $O(N^5)$ ．単 計算 $100^5 = 10^{10}$ ， 見5 内 答 思

応用例3：全探索の計算回数

N ー 5 選 ${}_N\mathrm{C}_5$ 通 ，$N = 100$ 合 ー 選

$$ {}_{100}\mathrm{C}_5 = \frac{100 \times 99 \times 98 \times 97 \times 96}{5 \times 4 \times 3 \times 2 \times 1} = 75287520 $$

通 ． 10^9 大 下回 ，5 内 実行 予測

．

課題

課題

- 課題　Moodle
- 課題　　　　　　　　題　解答　　　　　　　　作成
- 作成　　　　　　　　実行　　　題　　　動作　　　　　　　確認
- 動作確認　　　　　　,　　　　　　　　　(`filename.java`)　Moodle　提

提出期限は 11月18日(月) 20:00 まで