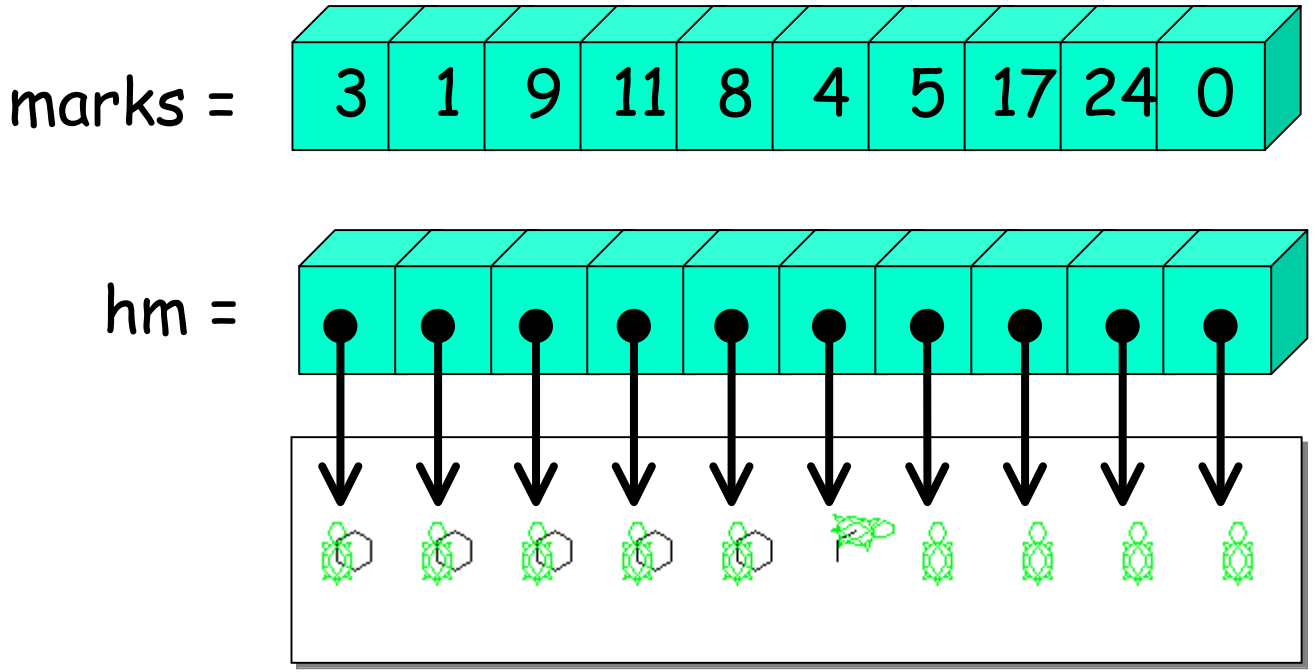
計算機プログラミングI

第5回　2002年11月7日(木)

- 配列: 沢山のデータをまとめたデータ
  - –どんなものか
  - –どうやって使うのか
  - –どういうときに使うのか
- Stringクラス
- mainメソッドの引数
- 配列を使ったアルゴリズム
- クイズ

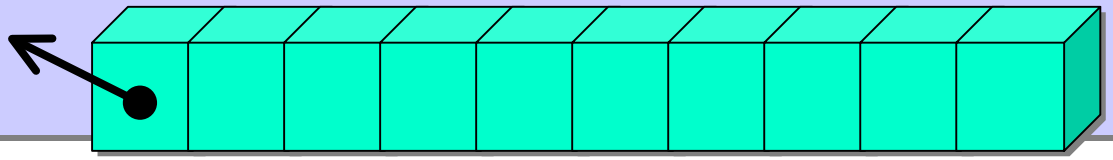
配列とは何か

- 変数の列を1つにまとめたオブジェクト
  (cf. ベクトル・数列)
- 番号で
  個々の
  変数を
  指定できる



配列の使い方

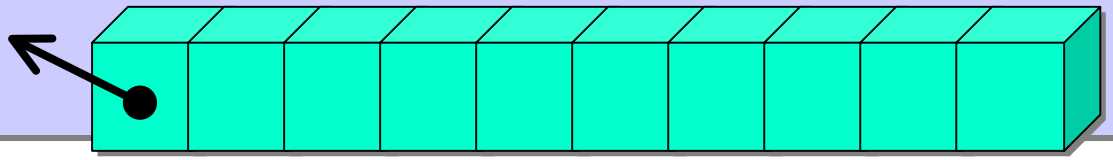
```
public class T51 {
  public static void main(String[] args){
    TurtleFrame f = new TurtleFrame(600,300);
    Turtle[ ] hm = new Turtle[10];
    for(int i = 0 ; i < 10; i++){
      hm[i] = new Turtle(i * 50 + 25,150,0);
      f.add(hm[i]);
    }
    //略
  }
}
```



配列の使い方

```
public class T51 {
  public static void main(String[] args){
    TurtleFrame f = new TurtleFrame(600
    Turtle[ ] hm = new Turtle[10];
    for(int i = 0 ; i < 10; i++){
      hm[i] = new Turtle(i * 50 + 25,150,0);
       f.add(hm[i]);
    }
    //略
  }
}
```

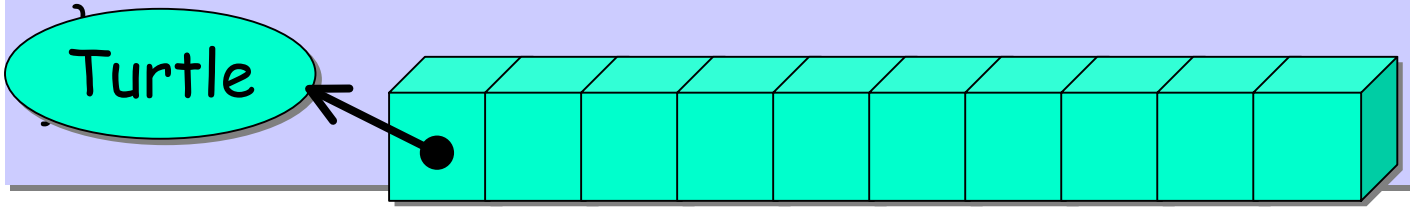
Turtleオブジェクトを
10個入れることのできる
配列を作る



配列の使い方

```
public class T51 {
  public static void main(String[] args){
    TurtleFrame f = new TurtleFrame(600
    Turtle[ ] hm = new Turtle[10];
    for(int i = 0 ; i < 10; i++){
      hm[i] = new Turtle(i * 50 + 25,150,0);
       f.add(hm[i]);
    }
    //略
```

Turtleオブジェクトを
10個入れることのできる
配列を作る

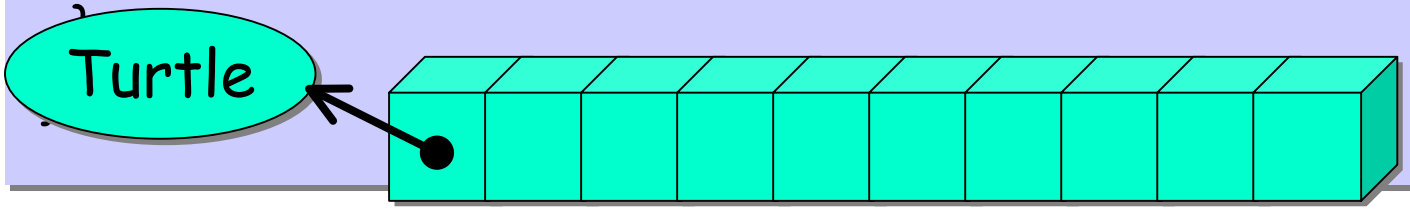


配列の使い方

```
public class T51 {
  public static void main(String[] args){
    TurtleFrame f = new TurtleFrame(600
    Turtle[ ] hm = new Turtle[10];
    for(int i = 0 ; i < 10; i++){
      hm[i] = new Turtle(i * 50 + 25,150,0);
      f.add(hm[i]);
    }
    //略
```

Turtleオブジェクトを10個入れることのできる配列を作る

配列のi番目



配列の使い方: 作る

```
int [ ] marks = new int [10];
```

new int [10] ——10個の整数をしまう配列を作る (中身は全て0)

new Turtle[10] ——10個のTurtleオブジェクトをしまう配列を作る (中身は全てnull)

int [ ] ——整数(int)をしまう配列の型

Turtle [ ] ——Turtleオブジェクトをしまう配列の型

配列の使い方: 作る

- new Turtle [ ] { new Turtle(), new Turtle() };
  - Turtleオブジェクトを入れる大きさ2の配列を作る
  - Turtleオブジェクトを2個作り、0番目、1番目にそれぞれ入れる

- Turtle [ ] hm; —— 配列をしまう変数を宣言
- hm = new Turtle [ ] { new Turtle(), new Turtle() };
  —— 配列を作り、代入

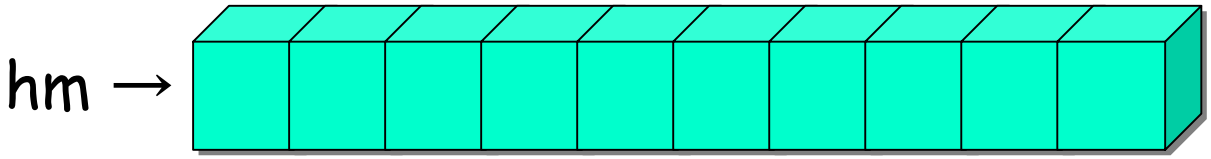
- Turtle [ ] hm = new Turtle [ ] { new Turtle(), new Turtle() };

- Turtle [ ] hm = { new Turtle(), new Turtle() };

配列の使い方: 使う

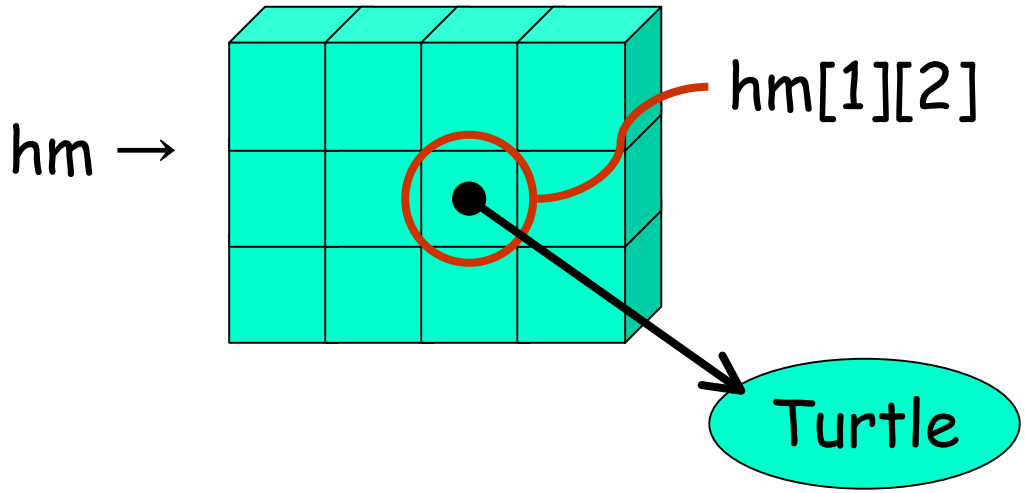
```
Turtle[ ] hm = ...;
for(int i = 0 ; i < hm.length; i++){
  hm[i] = new Turtle(i * 50 + 25,150,0);
  f.add(hm[i]);
}
```

- hm[i]　　配列のi番目　　(先頭は0番目)
  値をとり出す・代入する
- hm.length　配列の長さ



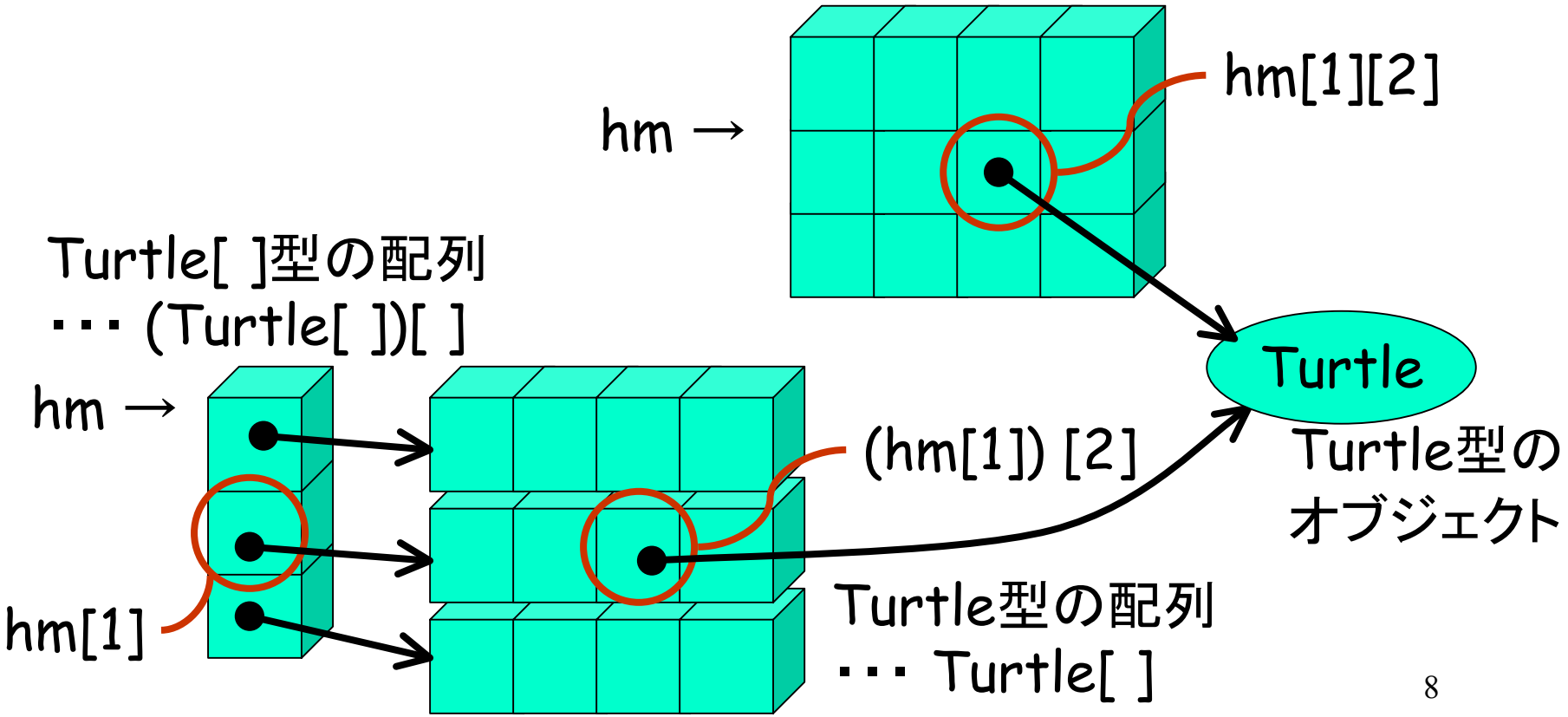
多次元の配列

- Turtle [ ] [ ] hm = new Turtle[3][4];



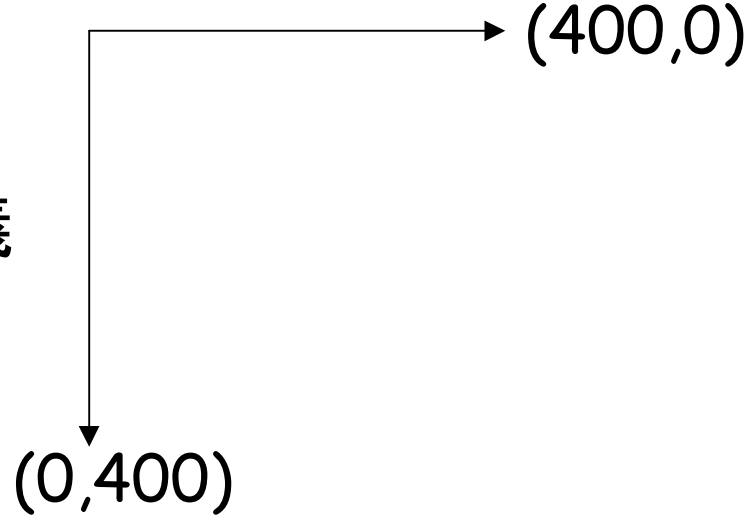
多次元の配列

・Turtle [ ] [ ] hm = new Turtle[3][4];



練習

- 5.3　初期値をもった配列を使って
  辺の長さを自由に変える
- 5.6　座標系に注意
- 5.8　5.6で「グラフを描く
  クラスメソッド」を定義
  しておくと簡単



Stringクラス・mainの引数

- 5.4 Stringクラス
  - 文字列を表わすオブジェクトのクラス
  - “あいうえお”　　—— Stringオブジェクトを作る
  - “x=” + x　　—— 文字列への変換・文字列の結合
- 5.5 mainの引数
  - public static void main(String[] args) {
  - java T21 red 10 として起動した場合
    args = new String [ ] { “red”, “10” } として実行される

配列を使ったアルゴリズム

- 素数
  - 単純
  - エラトステネスのふるい
- 組み合わせ数

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

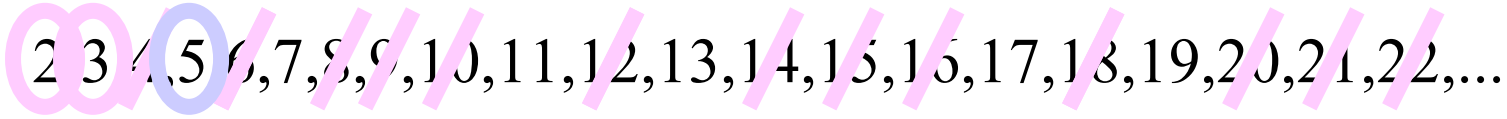
xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」



xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

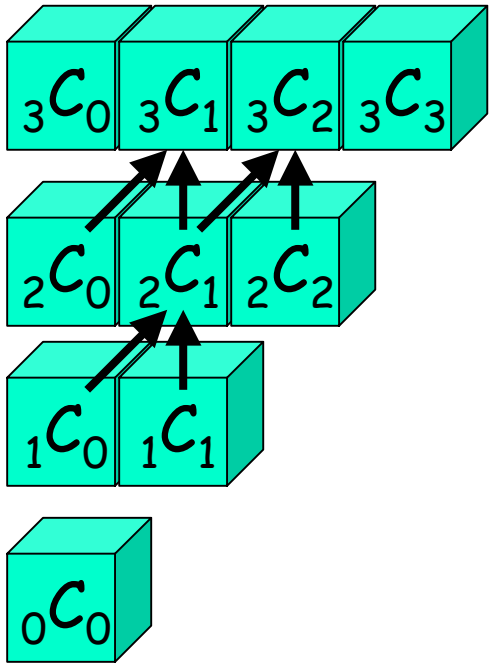
xが素数かを判定

- 単純なもの
  - xの最小の約数を見つける
  - それがxならば素数
- エラトステネスのふるい
  - x以下の全ての数が「素数かも知れない」
  - 最小の「素数かも知れない」数は素数
  - 素数の倍数は「素数でない」

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,...

組み合わせ数

- $_{m}C_{n} = \begin{cases} 1 & (n=0,\ n=m\text{のとき}) \\ {}_{m-1}C_{n} + {}_{m-1}C_{n-1} & (\text{それ以外}) \end{cases}$



練習

- Args51.javaを理解する
- 資料5.2, 5.5

- 教科書5.9-13