計算機プログラミングI

第6回　2002年11月14日(木)

- アルゴリズムと計算量
  - 平方根を計算するアルゴリズム
    - 1. 単純なもの
    - 2. 二分法
    - 練習
  - 計算量
    - その目的
    - 定義
    - 平方根のアルゴリズムの計算量
- 第1回課題の説明

平方根の計算

- ある数xの平方根をδ以内の誤差で求める
- 単純なアルゴリズム:
  - $a=0, \delta, 2\delta, 3\delta, ...$のように変化させ
  - a^2と x を比較し、$x < a^2$となったら終了
- $(a-\delta)^2 < x < a^2$なので a が x の平方根の近似値

- 練習
  - 6-1 動かしてみる
  - 6-2 xを大きくしたときの実行時間

平方根:単純なプログラム

- 単純なアルゴリズム:
  - a=0, δ , 2 δ , 3 δ , ...のように変化させ
  - a^2と x を比較し、$x < a^2$となったら終了

```
public static double linear(double x) {
        double a = 0;          // 平方根の推定値
        while (aの自乗がx未満か) {
            aをδだけ増やす
        }
        return a;
    }
```

平方根: 単純なアルゴリズムの問題

例: 10^6の平方根を誤差10^{-6}以内で求めよ

- 1回目: $a=0$, $a^2=0$ $< 10^6$
- 2回目: $a=10^{-6}$, $a^2=10^{-12}$ $< 10^6$
- 3回目: $a=2\times10^{-6}$, $a^2=4\times10^{-12}$ $< 10^6$

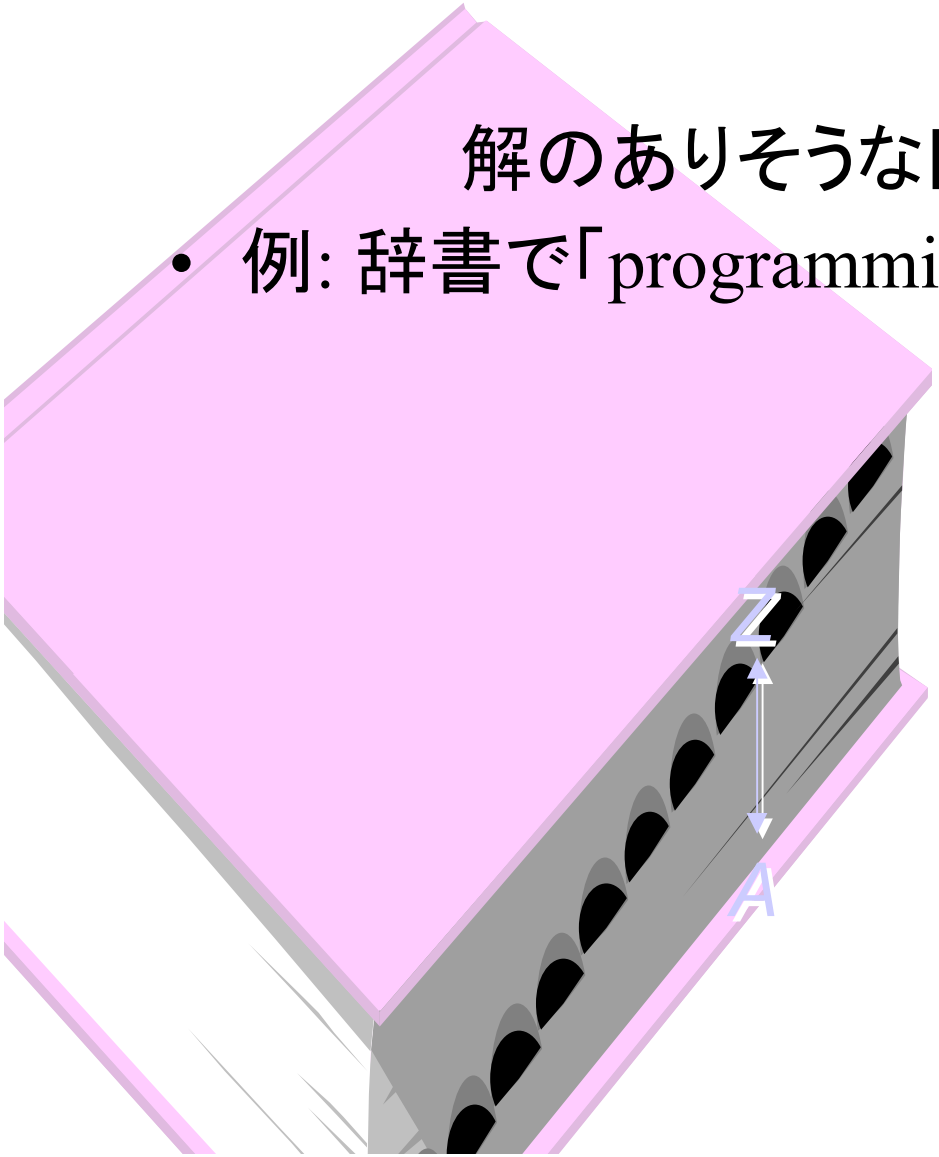
…

- cf. 辞書を1ページずつめくって探す
- 解から遠いところを細く探しても無駄

平方根: 二分法

解のありそうな区間を狭めてゆく

- 例: 辞書で「programming」を引く



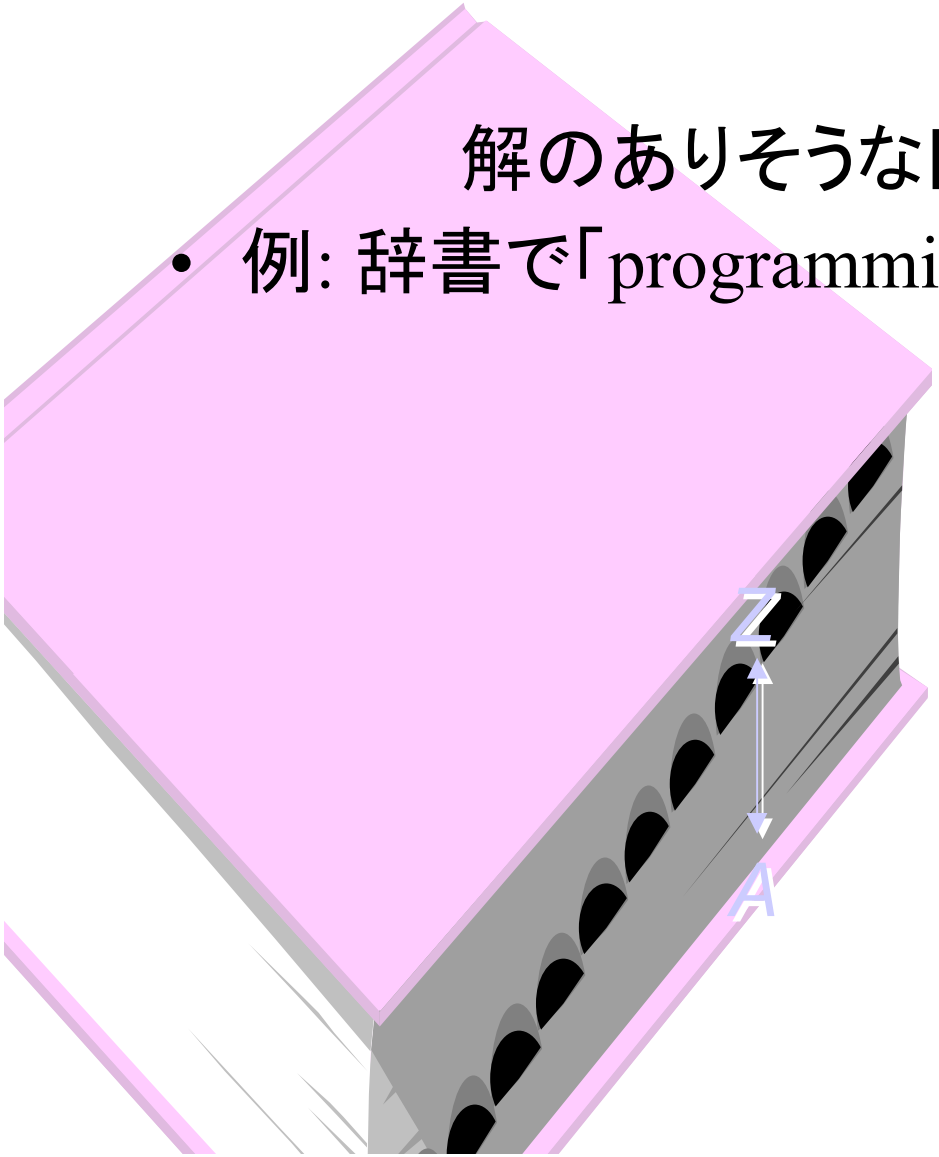
ポイント:

推測値から正解の
ある範囲が分かる

平方根: 二分法

解のありそうな区間を狭めてゆく

- 例: 辞書で「programming」を引く



learning

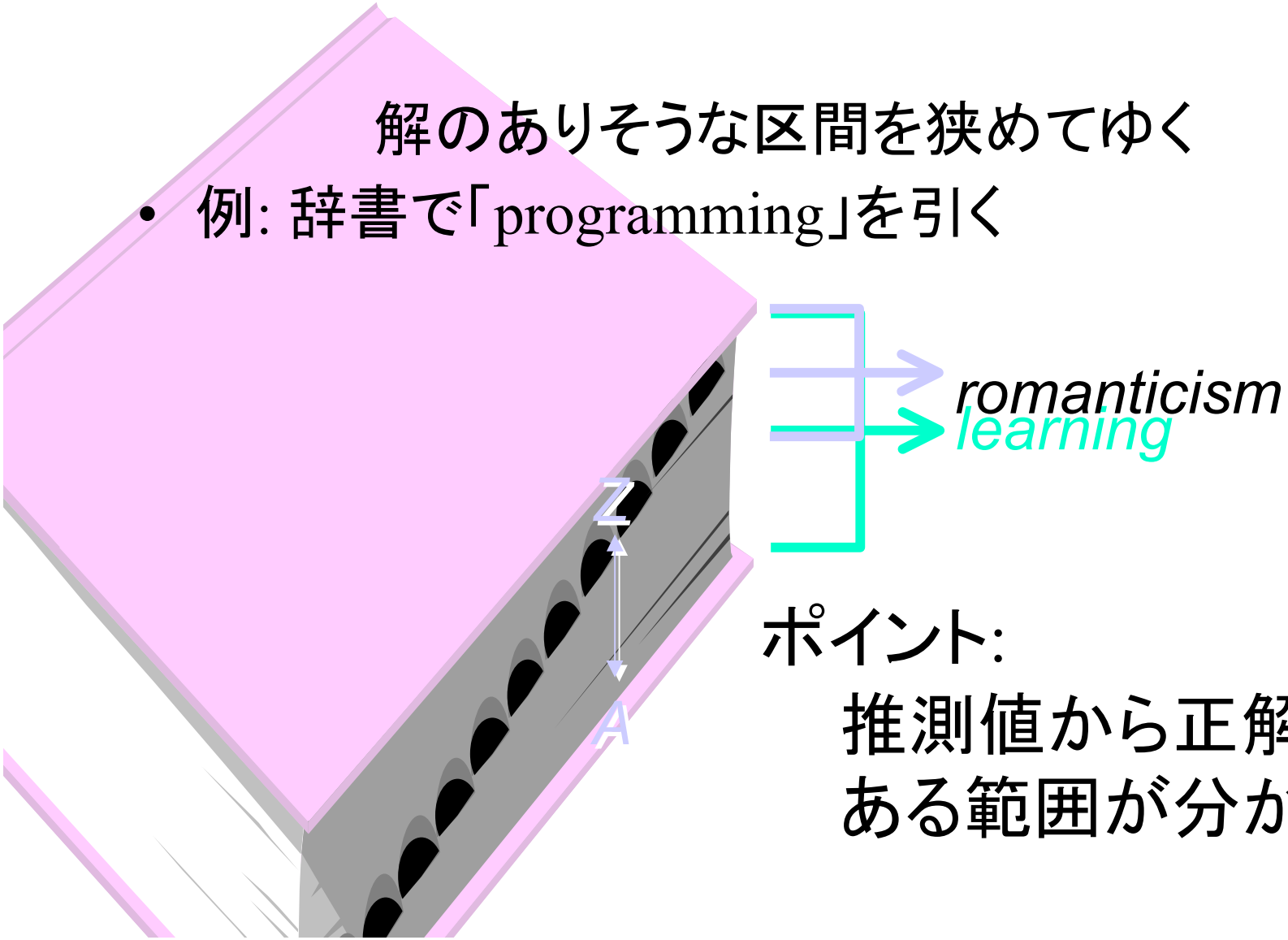
ポイント:

推測値から正解のある範囲が分かる

平方根: 二分法

解のありそうな区間を狭めてゆく

- 例: 辞書で「programming」を引く



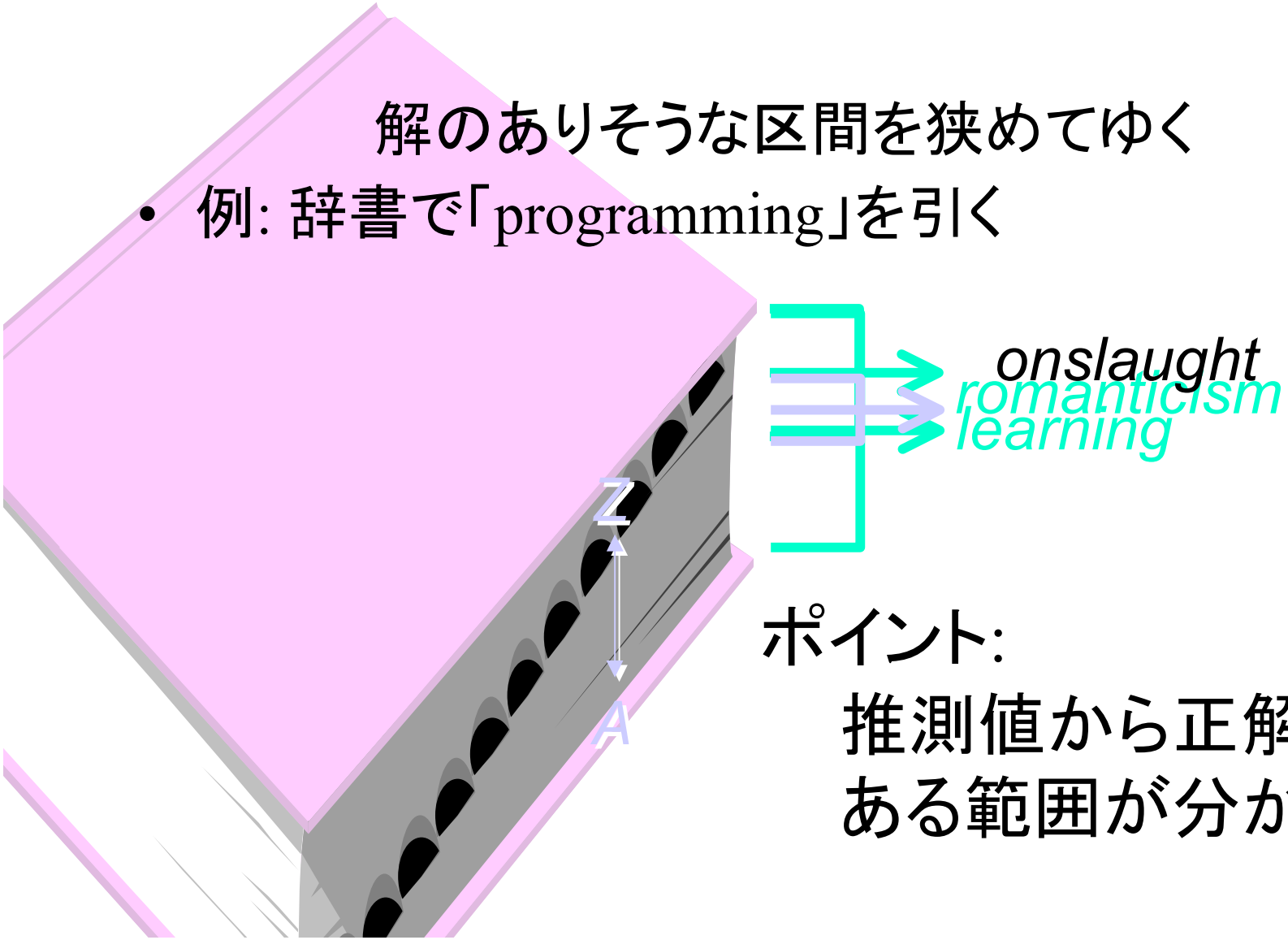
ポイント:

推測値から正解の
ある範囲が分かる

平方根: 二分法

解のありそうな区間を狭めてゆく

- 例: 辞書で「programming」を引く



ポイント:

推測値から正解の
ある範囲が分かる

平方根: 二分法

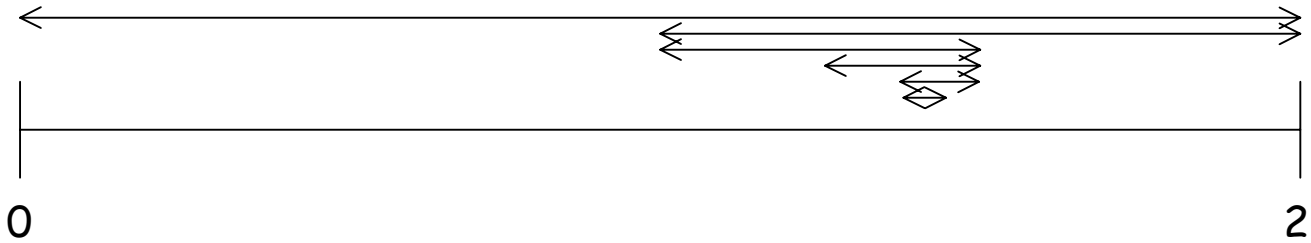
解のありそうな区間を狭めてゆく

- 例: 辞書で「programming」を引く

ポイント:

推測値から正解の
ある範囲が分かる

平方根: 二分法

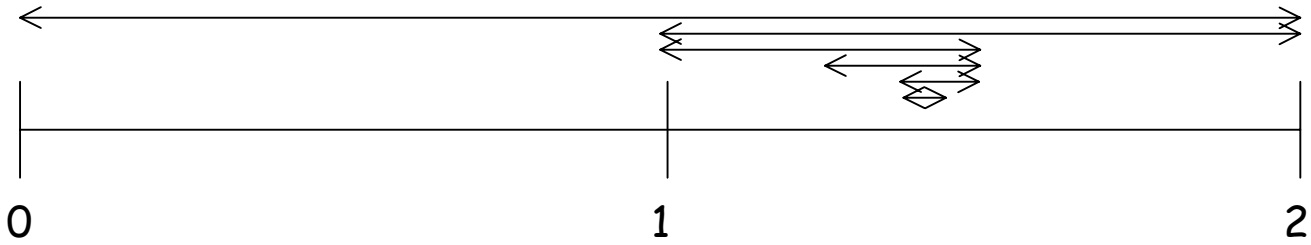


区間の幅はいずれ δ 以下に

2乗した値:

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法

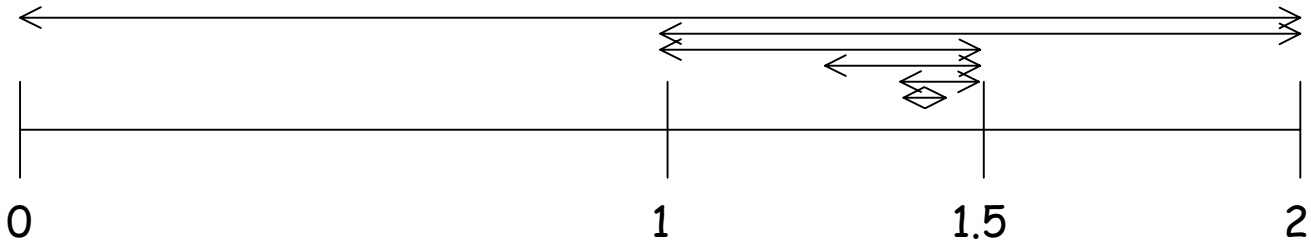


2乗した値: 1

区間の幅はいずれ δ 以下に

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法

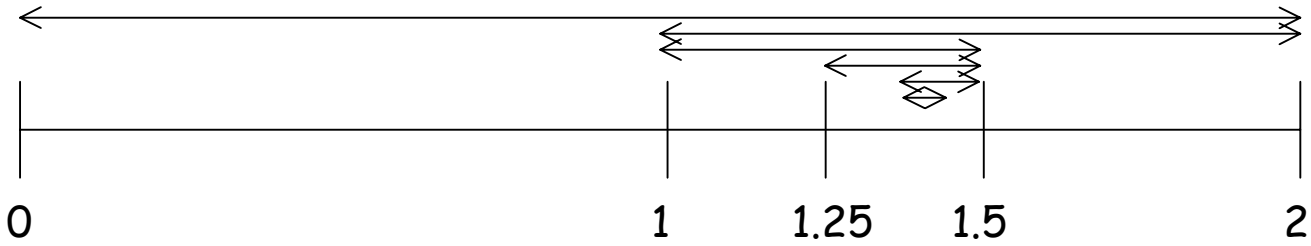


2乗した値: 1　　2.25

区間の幅はいずれ δ 以下に

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法

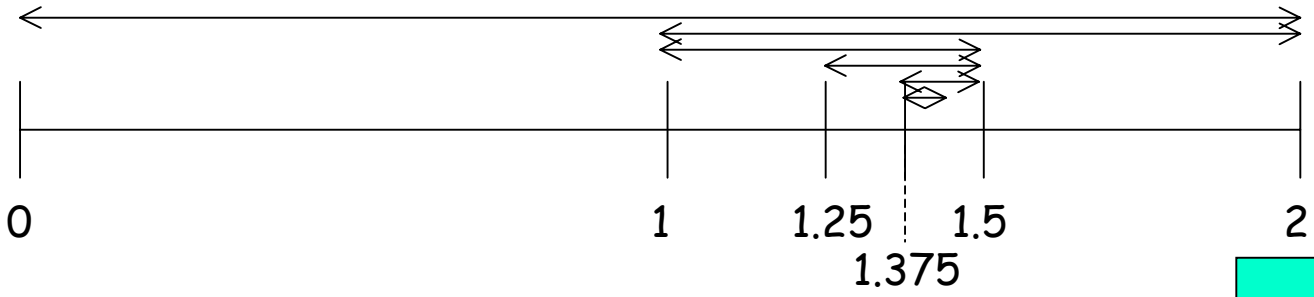


2乗した値: 1　　1.5625　　2.25

区間の幅はいずれ δ 以下に

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法

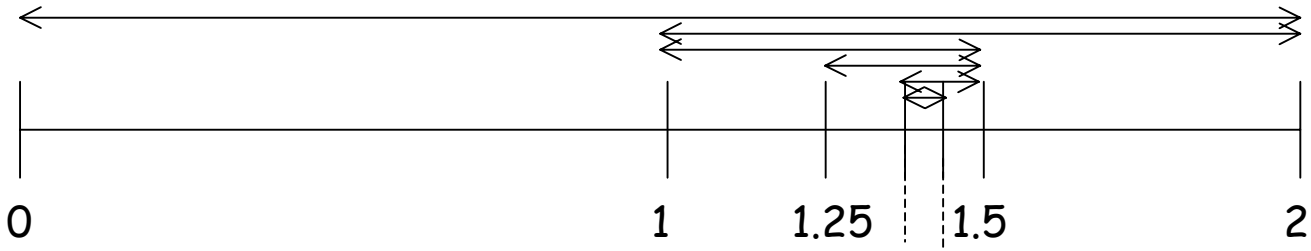


2乗した値: 1　1.5625　2.25
1.890625

区間の幅はいずれ δ 以下に

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法



2乗した値: 1　1.5625　2.25
1.890625
2.06640625

区間の幅はいずれ δ 以下に

回数	区間	中央値	中央値の 2 乗	(大小)	次の区間
1	(0 ,2)	1.0	1	(< 2)	(1 ,2)
2	(1 ,2)	1.5	2.25	(> 2)	(1.5 ,2)
3	(1.5 ,2)	1.25	1.5625	(< 2)	(1.25 ,1.5)
4	(1.25 ,1.5)	1.375	1.890625	(< 2)	(1.375,1.5)
5	(1.375,1.5)	1.4375	2.06640625	(> 2)	(1.375,1.4375)
		⋮			

平方根: 二分法

- min, max: 区間の下限・上限とする
- はじめ: min=0, max=x
- 区間の幅が δ 以下になるまでの間
  - 区間の中央値midの2乗と x を比較
    - $mid^2 < x$ のとき: 下限をmidまで上げる
    - $mid^2 > x$ のとき: 上限をmidまで下げる

平方根: 二分法のプログラム

- min, max: 区間の下限・上限とする
- はじめ: min=0, max=x
- 区間の幅が δ 以下になるまでの間
  - 区間の中央値midの2乗とxを比較
    - $mid^2 < x$ のとき: 下限をmidまで上げる
    - $mid^2 > x$ のとき: 上限をmidまで下げる

```
public static double binary(double x) {
  double min = 0, max = x; // 区間の下限・上限
  while (区間の幅はδ以上か) {
    if (区間の中央値の2乗がx未満か) {
      次の区間の下限を今の中央値にする
    } else {
      次の区間の上限を今の中央値にする
    }
  }
  return max;
}
```

平方根: 二分法の練習

- 6-3: 作る
- 6-4: 線形探索との比較
  - 値の正しさ
  - 計算時間
- 6-5: 平方根以外の関数

  $f(x)=\sqrt{x}$の場合　　a^2の大小をxと比較

  $f(x)=1/x$の場合　　axの大小を1と比較

  $f(x)=\sqrt[3]{x}$の場合　a^3の大小をxと比較

計算量

- 例: xの平方根を計算するアルゴリズム
  - 単純なもの　　$O(\sqrt{x})$
  - 二分法　　$O(\log x)$

——およその所要時間

- 定義:
  - n:問題の大きさ。
  - (時間計算量): あるプログラムがnの問題を解くのにかかる時間のうち、nに関係する主要な非定数項
  - $O(f(n))$ のように書く

計算量の例

- 1からnまでの和を求める
  - n回のくりかえし
  - 1回あたり足し算2回・比較が1回
  - 計$3n$回の計算 ——計算量はO(n)

```
int n = 12345;                  // 上限
int sum = 0;                    // 合計
for (i = 1; i < n; i++) {
  sum = sum + i;
}
```

計算量の例

- n個の値の標準偏差を計算する
  - n回のくりかえし——1回あたり5回の演算
  - その後5回の演算
  - 計$5n+5$回の計算　　——計算量は$O(n)$

```
int n = 12345;                    // 上限
int sumSquare = 0;                // 2乗和
int sum = 0;                      // 合計
for (i = 1; i < n; i++) {
  int m = mark[i];
  sum = sum + m;
  sumSquare = sumSquare + m*m;
}
double ave = sum/n;
double stdDev= Math.sqrt(sumSquare/n - ave*ave);
```

定数倍・定数項は無視する

計算量の例

- 単純な素数の個数の数え上げ
  - n回の繰返し——x回目は最大 $cx+d$ 回の演算

    演算回数の計 $(2c+d)+(3c+d)+(4c+d)+\ldots+(nc+d)$

    $= cn^2/2 + dn$

    定数およびnの項を無視して $O(n^2)$

```
int n = 100000;                    // 調べる最大値
int numPrimes = 0;                 // 素数の個数
for (int x = 2; x < n; x++) {
   int i = 2;                      // xの最小の約数を見つけて
   while (x % i != 0) {            // iに代入する
            i++;
   }
   if (x == i) {                   // x==iならxは素数
            numPrimes++;
   }
}
```

計算量の例: 平方根

- 問題の大きさ: x ($\sqrt{x}$を求める数)
- 単純なアルゴリズム

  $\sqrt{x}$ / δ 回の繰返し —— $O(\sqrt{x})$

```
public static double linear(double x) {
  double a = 0;// 平方根の推定値
  while (aの自乗がx未満か) {
    aをδだけ増やす
  }
  return a;
}
```

計算量の例: 平方根

- 問題の大きさ: x
  ($\sqrt{x}$を求める数)
- 二分法:
  - 繰返しの回数は?
    - 区間の幅に注目:
      最初 x, 2回目$x/2$,
      3回目$x/4$, 4回目$x/8$, ...
    - $x/2^k < \delta$ になったら終了、
      つまり約$\log_2(x)-\log_2(\delta)$回
  - 繰返し1回あたりの計算は
    定数

よって$O(\log x)$

```
public static double binary(double x) {
  double min = 0, max = x; // 区間の下限・上限
  while (区間の幅はδ以上か) {
    if (区間の中央値の2乗がx未満か) {
      次の区間の下限を今の中央値にする
    } else {
      次の区間の上限を今の中央値にする
    }
  }
  return max;
}
```

計算量の利用

- 平方根を求める時間 (秒)

x	1.E+00	1.E+02	1.E+04	1.E+06	1.E+08
単純	0.56	0.64	1.3	8.35	78.4
二分法	0.55	0.55	0.55	0.54	0.59

- 計算量の変化

x	10^0	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}
$\sqrt{x}$	1.0	3.2	10.0	31.6	100.0	316.2	1000.0	3162.3	10000.0	31622.8	100000.0
$\log x$	0.0	2.3	4.6	6.9	9.2	11.5	13.8	16.1	18.4	20.7	23.0

- O($\log x$)のアルゴリズムは重要

練習 (計算量)

- 6-6: 平方根のプログラムにおける計算量と実行時間の関係
- 6-7: 許容誤差も含めた計算量

  δ を小さくするとそれだけ時間も増えるはず
- 6-8: 冪乗のアルゴリズム

  $x^n = x*x*x*\cdots*x$　　掛け算をn回

  $x^{13} = x*x^{12} = x*(x^2)^6 = x*((x^2)^2)^3 = x*((x^2)^2)*((x^2)^2)^2$

  掛け算を5回

  計算量は?

第1回課題: Morphing

- 期限:　12月4日(水)
- 提出方法: 以下のどちらか
  - HTML形式で作ったファイルをスクリプト実行によって提出(詳細は追って通知する)、あるいは
  - 紙に書かれたものをレポートボックスへ提出
- 提出物: 各項目について、課題の考察・解いた方法の説明、コンパイル・実行可能なプログラム。
- 注意事項:
  - 常識の範囲を越えた類似部分のあるレポートがあった場合は、追加の面接試験を行う場合や、当該レポートの評価を0点にすることがある。
  - 全ての項目を提出しなくてもよい。
  - 複数の項目をまとめて解いた場合には、分かるように明記すること。
  - 提出物の中心は考察および解いた方法の説明である。(プログラムは説明が正しいことの証明に過ぎないので、それだけを提出してはならない。)
  - レポートの読みやすさ・独自性も採点の対象である。