基本情報技術 I

11. 探索処理, 計算量

菊池浩明

検索問題

- 問題
  - 次の表の金額から駅名を調べるプログラムを作れ.
  - 前提: キー（料金）は______されている.

料金	データ
150	新宿
160	四ツ谷
210	東京
290	国分寺
380	立川
450	日野
460	八王子
540	高尾

1. 線形探索

```
int linear(int k, int key[ ], String val[ ])
    for(int i = 0; i < key.length; ++i){
        if(key[i] == k){
            return(i);
        }
    }
    return(-1);
}
int key[ ] = new int[] {150, 160, 210};
String val[ ] = new String[ ]{"新宿", "四ツ谷", "東京“};
```

0	1	2	3	4	5	6	7
150	160	210	290	380	450	460	540

i

linear(210)=

linaer(300)=

線形探索の性能

- 平均比較回数

  k = 150　1回目　　　　確率1/8 (_____)

  k = 160　2回目　　　　確率1/8

  …

  k = 540　8回目　　　　確率1/8 (最悪)

  - 平均:
    (1+2+…+8)/8 = ((1+8)*8/2)/8 = (1+8)/2
    = ___________________
  - _____比較回数　　n 回

2. 二分探索★★頻出

```
int binarySearch(int q, int key[ ], String val [ ])
  int i = 0;  int j = key.length -1;
  while(j - i > 1){
     int k = floor((i + j)/2);
     if(key[k] == q){ return(k);
     }else if(q < key[k] ){ j = k;
     }else{ i = k; }
   }
   return(-1);
}
```

binary(210)=

binary(300)=

0	1	2	3	4	5	6	7
150	160	210	290	380	450	460	540

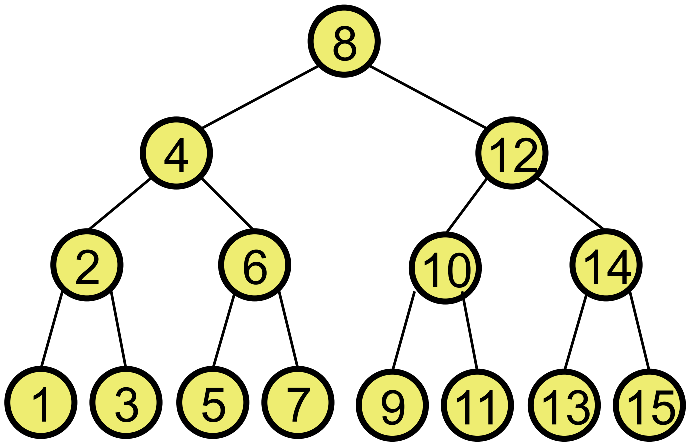
i=0　k=3　j=7

i=0　k=1　j=3

i=1　k=2　j=3

二分探索の性能

- ＿＿＿＿＿＿比較回数
  - ❑q = 290 k = n/2 1回目　最良
  - ❑q = 160 k = n/4 2回目　(q = 450も)
  - ❑q = 150 k = n/8 3回目　(準)最悪
  - ❑k回の比較の場合の数 n = 2^k : k = ＿＿＿＿＿＿



k=0　2^0

k=1　2^1

k=2　2^2

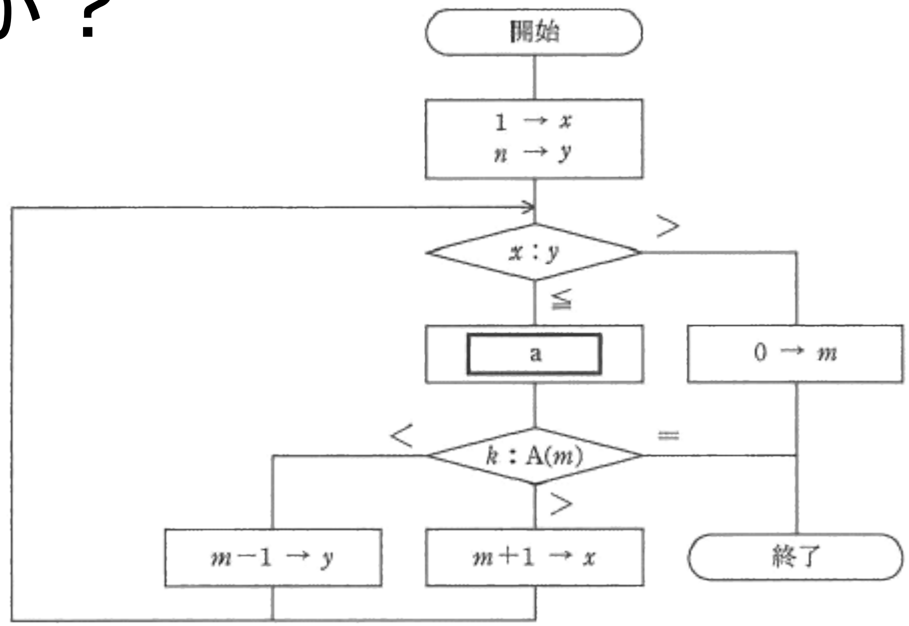
k=3　2^3

探索方式の比較

	線形検索		二分検索	
n	平均比較回数	処理時間	平均比較回数	処理時間
10	5	0.005	3	0.003
100	50	0.05	7	0.007
1000	500	0.5	10	0.010
10000	5000	5	13	0.013
100000	50000	50	17	0.017
1000000	500000	500	20	0.020
10000000	5000000	5000	23	0.023
100000000	50000000	50000	27	0.027
n	n		log(n)	

例1

- 整列済みの配列AからA[m]=kとなる要素mを二分探索法で見つける. aに入る式はどれか？



例2

- 2,000個の相異なる要素が昇順に整列された表がある. 2分探索して該当するキーを取出す. キーの比較回数は最大何回か？
  - ヒント: n=8の時, 3回である.

例3

- 2分探索において，データの個数が4倍になると，最大探索回数はどうなるか?

例4

- 次の文は正しいか.
  - 二分探索するデータ列は整列されている必要がある.
  - 二分探索は探索をデータ列の先頭から開始する.
  - n個のデータの探索に要する比較回数は n $\log_2 n$ に比例する.
  - 二分探索は線形探索より常に早く検索できる

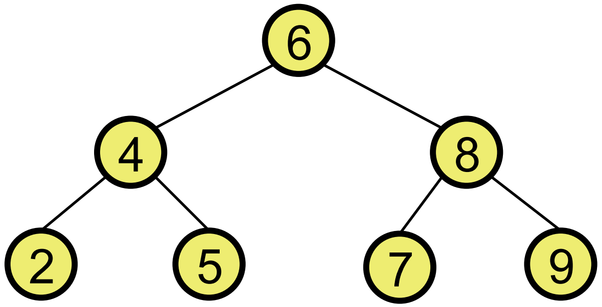
検索時間はデータに依存する

- search(150)

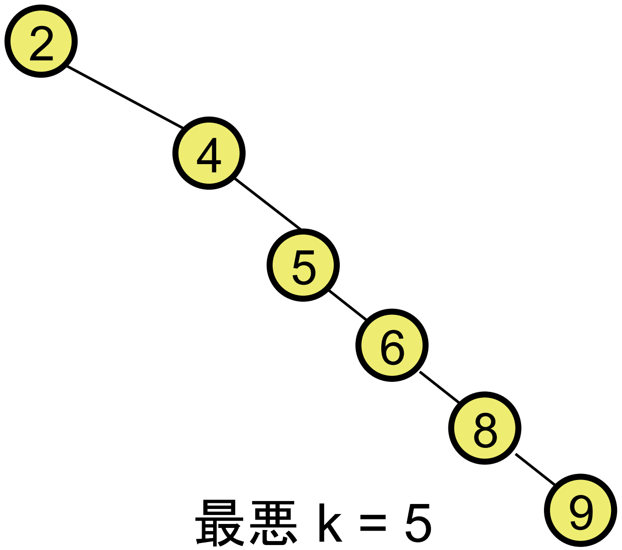
0	1	2	3	4	5	6	7
150	160	210	290	380	450	460	540

- 最良時
  - 線形検索 ____回でHIT
- 最悪時
  - 二分探索___回でヒット

- クイックソート



最良 k = 2



最悪 k = 5

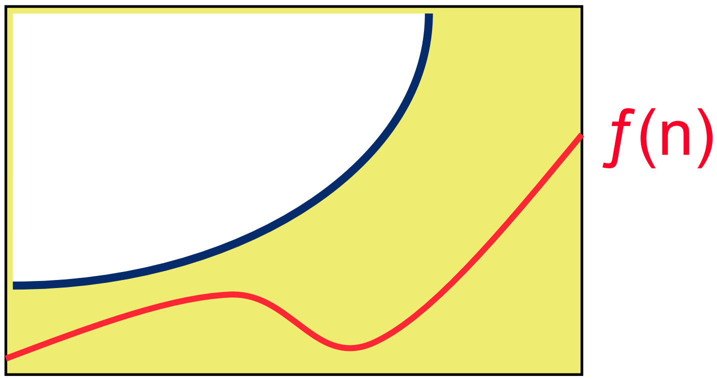
3. 計算量

- 計算量 O(n)
  - ❑ $O(f(n)) = f(n)$オーダで実行時間が増加するアルゴリズムの集合「＿＿＿＿＿＿」
  - ❑ $O(f(n))$ 漸近的＿＿＿ (最悪でも $f(n)$)
  - ❑ $\Theta(f(n))$ 漸近的タイトな限界
  - ❑ $\Omega(f(n))$ 漸近的下界 (最良では $f(n)$)

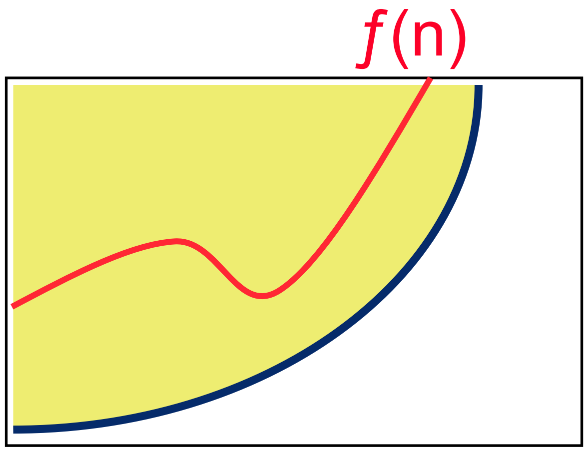
- 例
  - ❑ Insert Sort: $T_I(n) = \Theta(n^2)$; $T_I(n) \in \Theta(n^2)$
    $T_I(n) = \Omega(n)$
  - ❑ Merge Sort: $T_M(n) = \Theta(n \lg n)$

上界と下界

- 漸近的上界
  - $O(n^2) = \{ f(n) \mid$ すべての$n \ge n_0$に対して, $0 \le f(n) \le c\, n^2$となるc, n_0が存在する$\}$



- 漸近的下界
  - $\Omega(n^2) = \{ f(n) \mid$ すべての$n \ge n_0$に対して, $0 \le c\, n^2 \le f(n)$ となるc,n_0が存在する$\}$



計算量の求め方

- 実行時間(基本演算の回数）
  - $T(n) = n^2 \qquad = \Theta(n^2)$
  - $T(n) = 3\,n^2 - 2\,n + 6 \quad = \Theta(\quad)$
  - $T(n) = 100\,n^2 + 3 \quad = \Theta(\quad)$
- 厳密な定義
  - $\Theta(n^2)$ = { $f(n)$ | すべての$n \ge n_0$に対して,
    $0 \le c_1 n^2 \le f(n) \le c_2\, n^2$となる$c_1, c_2, n_0$が存在する}
  - 例）$T(n) = 3\,n^2 - 2\,n + 6$
    $(\quad)\,n^2 \le 3\,n^2 - 2\,n + 6 \le (\quad)\,n^2$ for all $n \ge 3 = n_0$

例5

- 探索方法とその実行時間を求めよ．ここで，探索するデータ数をnとする．

二分探索	線形探索	ハッシュ探索
	n	

例6

- 次のオーダーを小さい順に並べよ.
  - $n, n^2, n^3, 2^n, n!, \log n, n*\log n, \sqrt{n}$
  - ヒント: $n = 64$で比較せよ.

4. 再帰アルゴリズム★頻出

- 再帰呼出し
  - 関数が処理中に自分自身を呼び出すこと.
  - 例） 階乗 N! = N*(N-1)*….*1 = N*(N-1)!

```
int fact(int n){
    if(n == 0) return 1;
    return n*fact(n-1);
}
```

  - fact(5) =

公約数 common divisor ★頻出

- 公約数 d
  - aの約数でかつbの約数である数d
- 最大公約数 gcd(________________)
  - gcd(a,b): aとbの最大の公約数
  - a = 30 = 3 • 5 • 2, b = 12 = 3 • 2 • 2
    gcd(a,b) = 3 • 2 = 6
  - gcd(18,6) =
  - gcd(-30,12)=   (約数は正)
  - gcd(12, 0)=   (定義.gcd(0,0) = 0)

gcdの求め方 Euclid

- Euclid（____________の互除法）
  - ❑ 原典 B.C. 300年.「最古のアルゴリズム」
  - ❑ 再帰(recursive)

```
1.  static int gcd(int a, int b){
2.          if(b == 0) return a;
3.          return gcd(b, a % b);
4.  }
```

  - ❑ gcd(30, 21) = gcd(21, 30 mod 21)
    = gcd(9, 21 mod 9)
    = gcd(3, 9 mod 3) = ___
  - ❑ 再帰呼び出し回数 k = ___

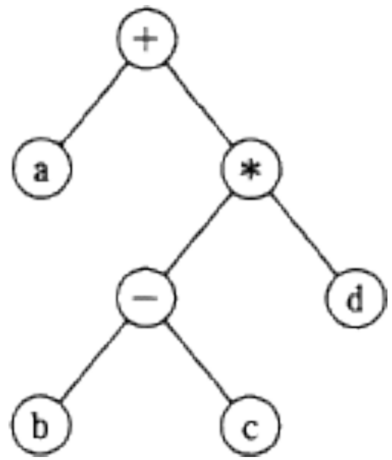
例7

- 次の関数f(x,y)について，f(775, 527)を求めよ.
  - ❑f(x, y): if y = 0 then return x else return f(y, x mod y)

例8

- 二分木の各ノードを出力する次のプログラムがある. これを図の二分木の根に適用した結果を求めよ.

1. proc(ノード n){
2. 　nに左の子*l* があれば proc(*l*)を呼び出す
3. 　nに右の子*r* があれば proc(r)を呼び出す
4. 　nの記号を出力する
5. }



例9

- 次の関数f(n, k)について，f(4,2)を求めよ.

$$f(n,\ k) = \begin{cases} 1 & (k = 0), \\ f(n-1,\ k-1) + f(n-1,\ k) & (0 < k < n), \\ 1 & (k = n). \end{cases}$$

まとめ

- 線形探索の(　　　　　)回数はn/2, 二分探索は(　　　　　)である.
- アルゴリズムの性能は入力サイズnに対する平均処理時間の増え方である(　　　　　)で計り，単位は(　　　　　)と呼ぶ.
- 関数が自分自身を呼出すことを(　　　　　)といい，代表例に階乗n!や(　　　　　)ソートやGCDを求める(　　　　　)がある.