

## アルゴリズムとデータ構造

- 「アルゴリズム」(algorithm) とは

- 問題を解く手順 . 日本語訳：算法，解法

- 基本的なアルゴリズムの種類は，ほんのわずか：探索 (search) と整列 (sorting)

探索：二分探索

整列：バブルソート，マージソート，クイックソート，...



図 10.3 数を大小順に並べる



図 10.4 最小値を探して仕切りの左に置く

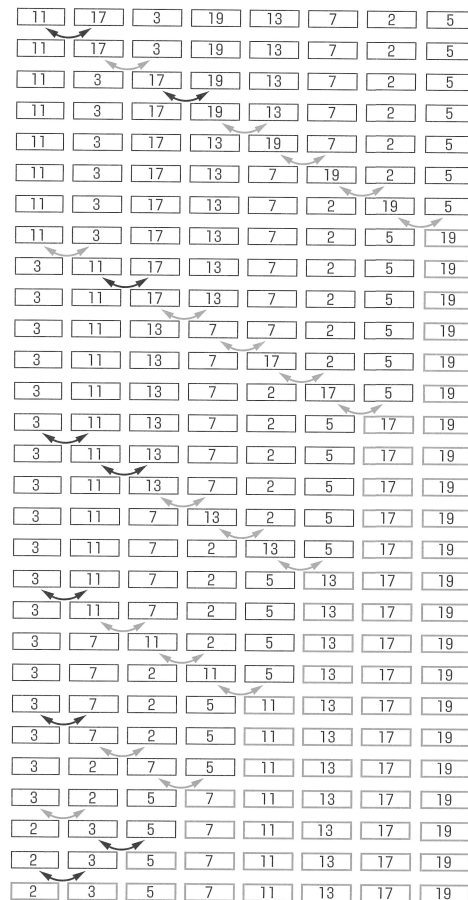


図 10.5 バブルソート

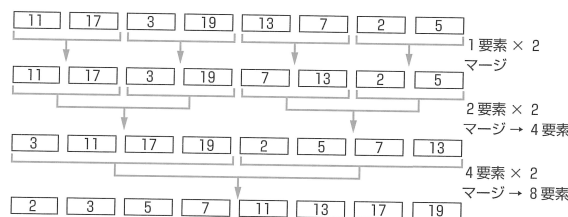


図 10.7 マージソート

- エレガントなテクニックと高度な数学的分析が満ち溢れている

普通なら何年もかかるような問題 ⇒ あっという間に解決

「アルゴリズムを工夫する」⇒ 情報処理の能率が大きく変化 (計算時間，記憶量)

- 「データ構造」(data structure) とは

- データを蓄えるための構造
- 配列 (array), リスト, 木 (tree), ハッシュテーブル (hash table)  
 配列: 静的にデータを記憶  
 リスト, ツリー, ハッシュテーブル: 動的にデータを記憶  
 「データ構造を工夫する」⇒ 情報処理の能率が大きく変化 (計算時間, 記憶量)
- 要素の追加, 削除に要する手間. ランダムアクセスする時間.

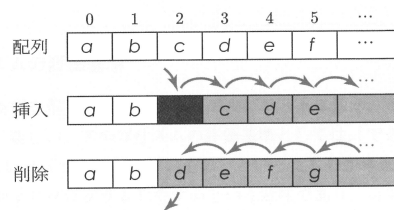


図 2.1 配列による列の表現

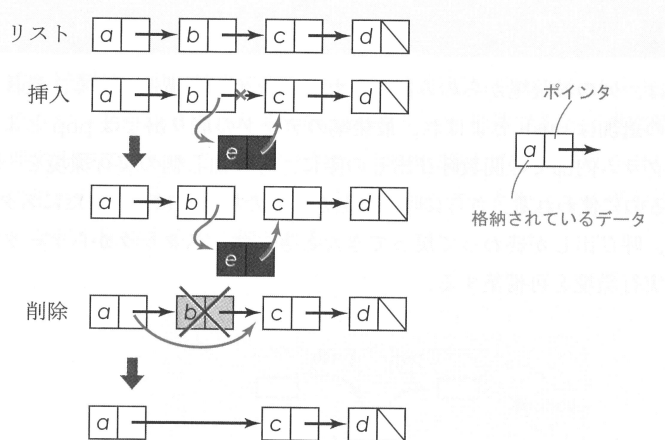


図 2.2 リストによる列の表現

- プログラムを書く

1. どんなデータ構造を採用するか決める
  2. どんなアルゴリズムを採用するか決める
  3. プログラムを書く. プログラム = アルゴリズム + データ構造 (?)
- 気をつける点: コンピュータは融通が効かない (プログラムに書いた通りにしか動かない)

- 例

- 例題 1. 変数 a と変数 b の値を入れ替えよ .
- 例題 2. 変数 x の 16 乗を計算せよ .  
 \* 計算の仕方 1 :  $x \times x \times \dots \times x = x^{16} \Rightarrow$  掛け算の回数 : 15 回 .  
 \* 計算の仕方 2 :  $x^2 = x \times x, x^4 = x^2 \times x^2, x^8 = x^4 \times x^4,$   
 $x^{16} = x^8 \times x^8 \Rightarrow$  掛け算の回数 : 4 回 .

問題：  $n$  個の整数  $(1, 2, \dots, n)$  から  $m$  個の重複のない乱数列を生成せよ。

例：  $n = 20, m = 10$  の場合。

出力例 1: 3 4 8 11 12 13 16 18 19 20

出力例 2: 1 2 8 9 10 11 12 13 17 20

出力例 3: 3 6 7 11 12 13 15 17 18

上記の目的を実現するプログラム：Robert W. Floyd 教授 (1936-2001) のアルゴリズム。

---

```
#include <stdio.h>
#include <stdlib.h>

int main(void){

    srand48(12388); /* 12388 は乱数の種 */

    int s[100]; /* 配列の大きさは n+1 以上に設定すること */
    int i,j,t;
    int m = 10;
    int n = 20;

    for (i=0; i < n+1; i++){
        s[i] = 0;
    }

    for (j = n-m+1; j <= n; j++){
        t = 1 + (int)( drand48()*(double)j ); /* 乱数を 1 つ生成し t に代入
                                             (t は 1 t j を満たす整数) */

        if (s[t] == 1 ){
            s[j] = 1;
        }
        else{
            s[t] = 1;
        }
    }

    for (i=0; i <n+1; i++){
        if ( s[i] > 0 ){
            printf("%d ",i);
        }
    }
    printf("\n");

    return 0;
}
```

---