

## 3. シェルソート

実践プログラミング I (3J)

情報工学科 鈴木雅人

# ソート (並び替え) とは

---

- ▶ ソート

大量のデータをある基準に従って並べ替えること

- ▶ ソートのアルゴリズム (方法・手順)

バブルソート

選択ソート

挿入ソート

シェーカーソート

シェルソート

クイックソート

マージソート

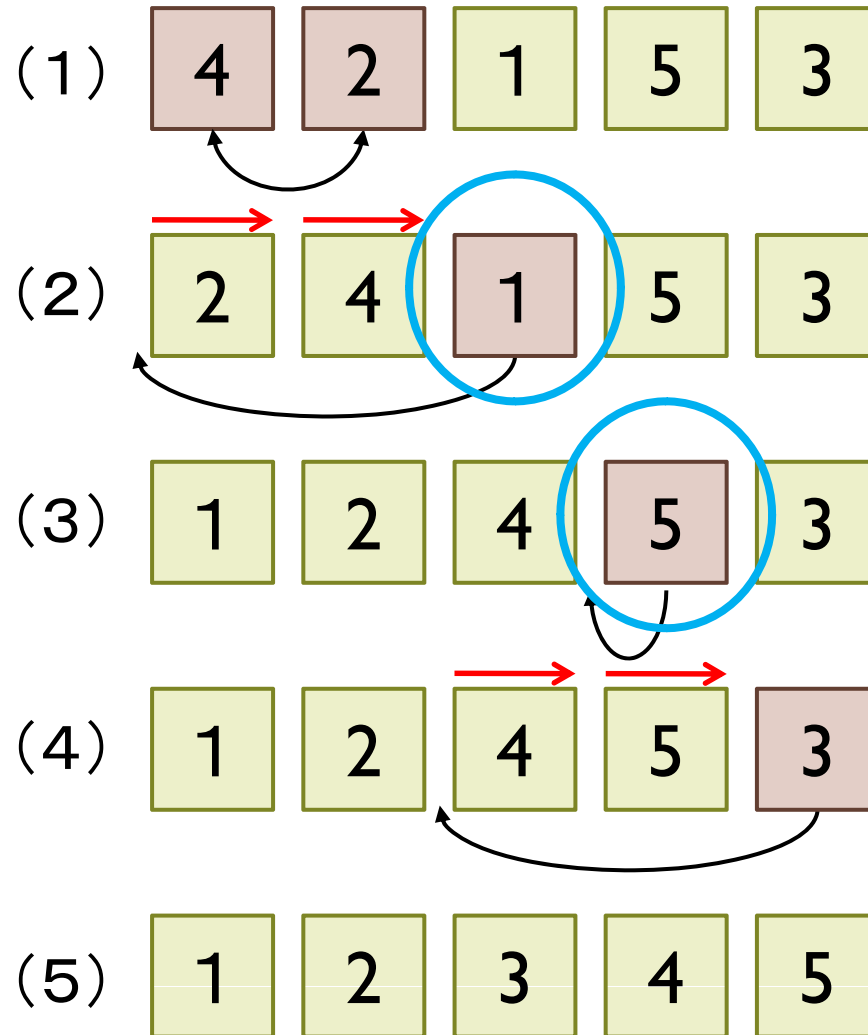
ヒープソート

} 単純ソート

} 単純ソートの改良



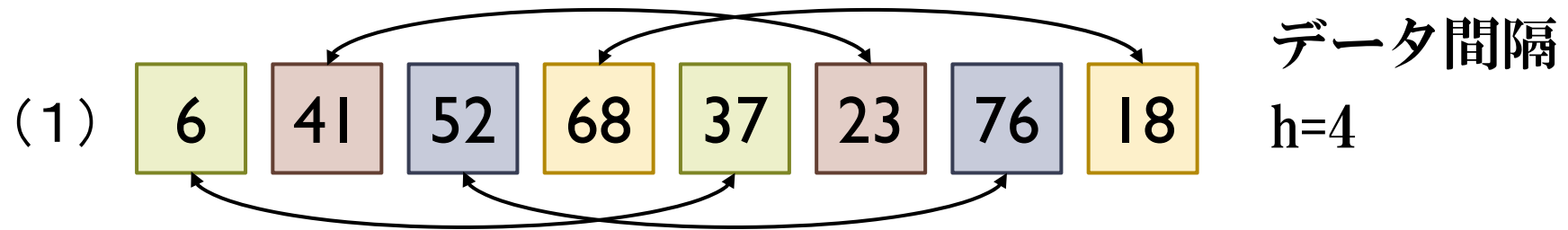
# 挿入ソートの問題点



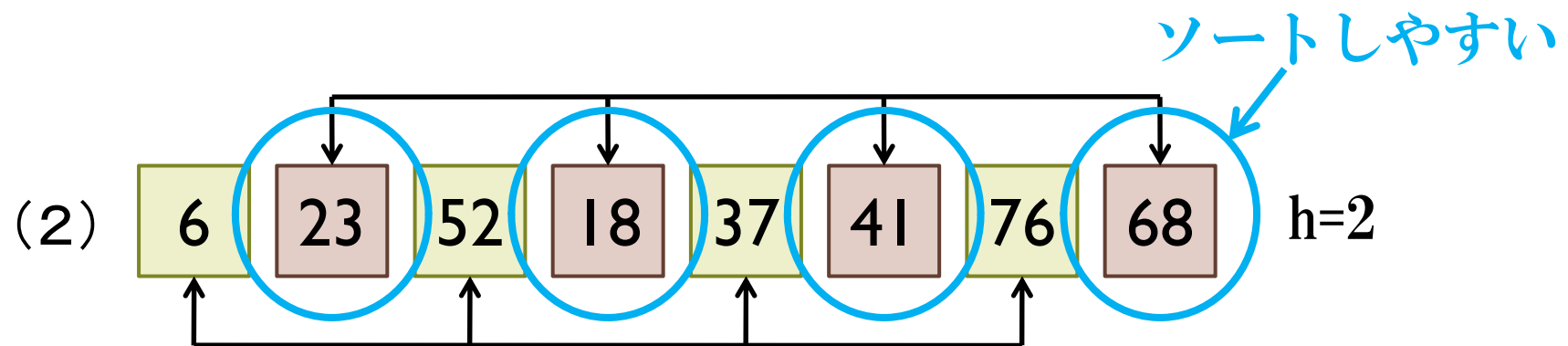
- ① データを横一列に並べる
- ② 左端の2つのデータをソートする
- ③ 3つ目のデータをソート済みの適切な位置に挿入し、他のデータを右にずらす
- ④ ③と同様の操作を4つ目以降のデータについて順に行う

データがある程度整列  
していればデータの移動  
回数は抑えられるはず

## シェルソートの手順



(6,37) , (41,23) , (52,76) , (68,18) という4つのデータ列をそれぞれ挿入ソートする



(6,52,37,76) , (23,18,41,68) という2つのデータ列をそれぞれ挿入ソートする

## シェルソートの手順

---

(3) 

6	18	37	23	52	41	76	68
---	----	----	----	----	----	----	----

 データ間隔  
h=1

改めて挿入ソートを行う

(4) 

6	18	23	37	41	52	68	76
---	----	----	----	----	----	----	----

 完成



# シェルソートのアルゴリズム

---

- データ間隔 $h$ のソートを「 $h$ ソート」と呼ぶ
- データ間隔 $h$ は、 $h=4,2,1$ のように取る必要はない
- 【 $h$ の一般形】  $h = m * h + n$   
(例)  $m=2, n=0$ の場合  
最後は $h=1$ となるので、 $h$ は $1,2,4,8,\dots$ となる
- 一般に $x > y$ のとき、 $x$ ソートを行った後に $y$ ソートを行うと、その結果は $x$ ソート済みかつ $y$ ソート済みになる

$h$ の数列は、減少して最後に1になりさえすれば、  
どのような数列でもかまわない。



# シェルソートのアルゴリズム

---

「 $h = 3 * h + 1$ 」で数列を生成するときのアルゴリズム

【小さい順に並べる場合】

- ① データを横一列に並べる
- ②  $h=1$ から始めて「3倍して1を加える」操作を繰り返し、データ数 $n$ を超えない最大数 $h$ を求める
- ③  $h$ ソート (挿入ソート) を行う
- ④  $h=1$ なら終了. そうでない場合は $h=h/3$ とし③を繰り返す



## 【課題3-1】

---

ファイルに書き込まれている100件のデータを読み込み、シェルソートを用いてそれらを小さい順に並べ替え、結果を画面に出力するプログラムを作成しなさい。

### 【注意】

サンプルデータは第1回課題のものを使うこと。

「 $h = 3 * h + 1$ 」で数列を生成してソートすること。





## 【課題3-2】（発展課題）

---

課題1-5のプログラムと同じものを，シェルソートを用いて作成しなさい。

また，同じデータをシェルソートと挿入ソートとでソートし，その速さの違いを調べなさい。

【注】 データ数が少ないと差が出ません



## 【課題3-3】（発展課題）

---

挿入ソートとシェルソートとを比べたとき、シェルソートの方が処理速度が速いことが理論的に証明されている。そのことを調査し、レポートにまとめなさい。

