

# プログラミング応用

<http://bit.ly/ouyou3d>

## アルゴリズム (3)

後期 第1週

2018/9/20

# 本日の目標

ソートイングアルゴリズム  
について理解します

コンピュータは、  
問題を解決するため（目的を達成するため）に、  
**手順に従って計算をする**



**問題**



**手順に従った計算**



**解決！**

**手順 = アルゴリズム**

# ソートイングとは？

大きさが異なる数字を…

小さい順（大きい順）に並び替える

6 2 9 1 5



1 2 5 6 9

問題



解決！



手順に従った計算

ソートイングアルゴリズム

# ソーティングとは？

世の中で一番よく使われている  
身近なアルゴリズム

例：商品を検索 → 安い順に並べる

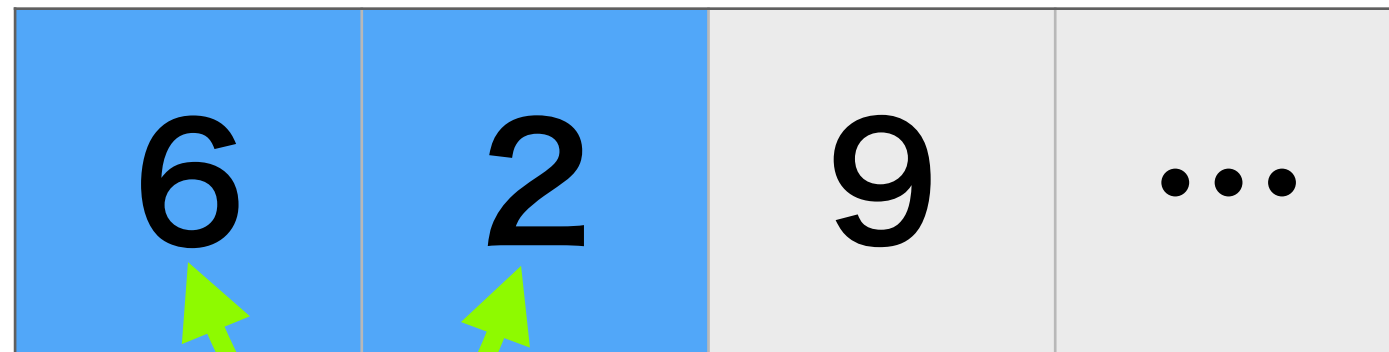
写真を整理 → 日付順に並べる

書類回収 → 出席番号順に並べる

ソーティングで  
最も単純なアルゴリズム  
バブルソート

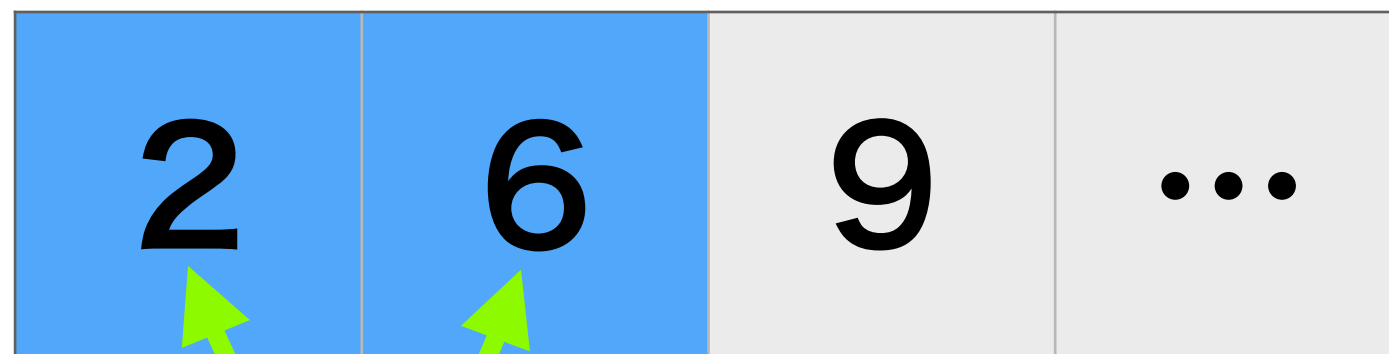
キーワードは、比較と交換

隣りあう数字を**比較**して…



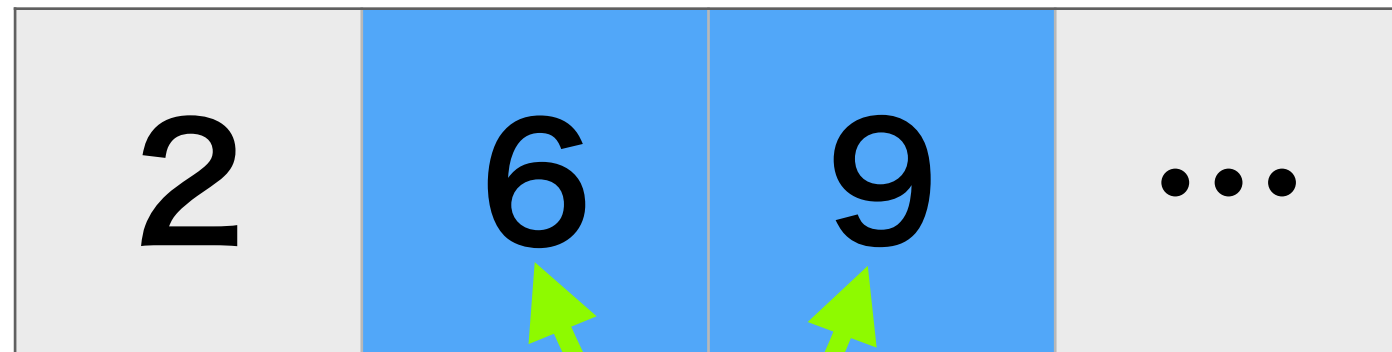
**どっちが大きい？**

右の方が大きくなるようにする



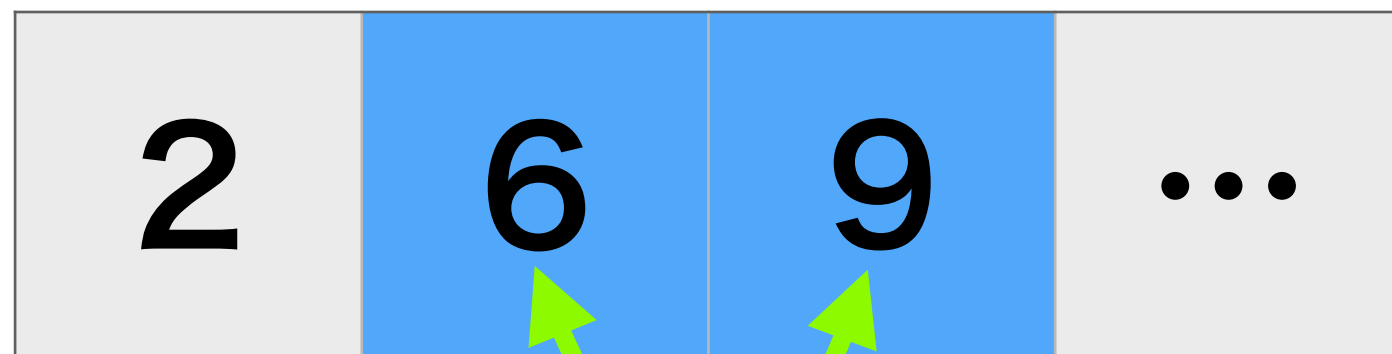
**6の方が大きい→交換**

さらに、隣りあう数字を**比較**して…



**どっちが大きい？**

右の方が大きくなるようにする



**9の方が大きい→交換しない**



# バブルソートの様子

7	2	1	6
---	---	---	---

2	1	6	7
---	---	---	---

1	2	6	7
---	---	---	---

1	2	6	7
---	---	---	---

1	2	6	7
---	---	---	---

黒色の数字の最大値が  
一番右側に  
浮き上がってくる  
↑ 泡 (バブル)

# バブルソートの様子

7	2	1	6
---	---	---	---

2	1	6	7
---	---	---	---

1	2	6	7
---	---	---	---

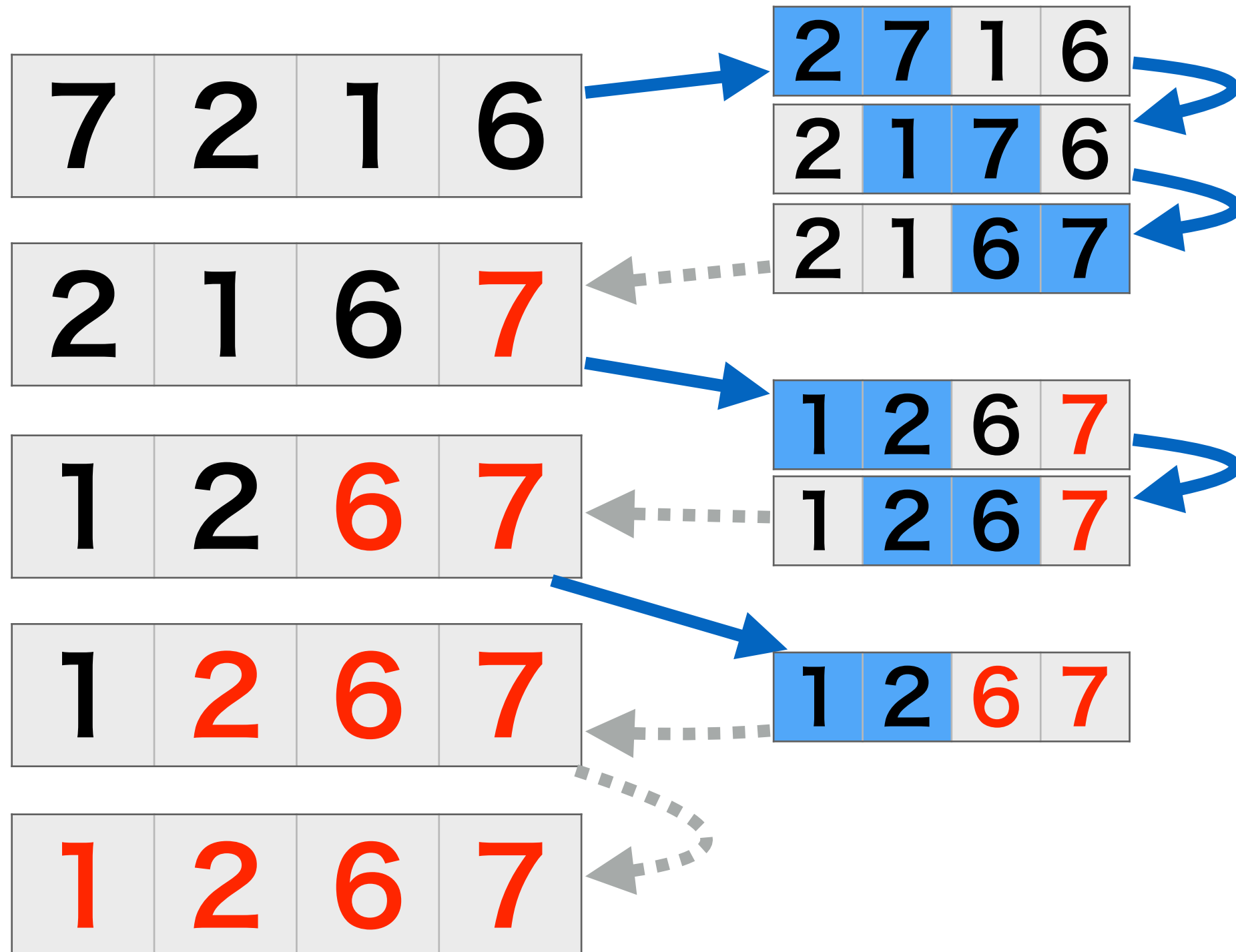
1	2	6	7
---	---	---	---

1	2	6	7
---	---	---	---

黒色の数字の最大値が  
一番右側に  
浮き上がってくる  
↑ 泡（バブル）

全ての数字が  
ソート済み  
になるまで繰り返す

→ は、比較と交換の処理  
..... は、最大値を赤字にする処理



# 今日の課題は

課題1～3で、バブルソートをする  
プログラムを少しずつ作っていきます

# 【課題1-1】

引数で与えられたint型配列の全要素の値を出力する関数show()を作成して下さい。

[この関数のプロトタイプ宣言]

```
void show(int value[], int size);
```

```
/* 配列valueの要素数はsizeで与えられる */
```

```
/* valueの0番目からsize-1番目までの値を出力する処理を繰り返す */
```

# 【課題1-1】

[mainの処理]

```
int v[10] = {9, 7, 6, 0, 1, 5, 2, 3, 4, 8};  
show(v, 10);
```

[実行結果]

```
9 7 6 0 1 5 2 3 4 8
```

# 【課題1-2】

引数で与えられた2つの整数の値を交換する関数  
`swap()`を作成して下さい。

[この関数のプロトタイプ宣言]

```
void swap(int *x, int *y);
```

```
/* ポインタxとyの参照先の値を交換する */
```

```
/* プログラミング基礎の後期第3週でも同じ関数を作っている */
```

# 【課題1-2】

[mainの処理]

```
int a = 10;  
int b = 20;  
printf(" (交換前) a: %d, b: %d\n", a, b);  
  
swap(&a, &b);  
printf(" (交換後) a: %d, b: %d\n", a, b);
```

[実行結果]

(交換前) a: 10, b: 20

(交換後) a: 20, b: 10



# 【課題1-3】

引数で与えられたint型配列の値をバブルソートする関数bubble()を作成して下さい。

[この関数のプロトタイプ宣言]

```
void bubble(int value[], int size);
```

```
/* 配列valueの要素数はsizeで与えられる */
```

```
/* 先に示したイラストと以下のアルゴリズムを参考に作る */
```

先のイラストをアルゴリズムでまとめると・・・

- ▶ 変数jの繰り返し回数を表す変数nを用意して、初期値はsize-1にする
- ▶ 変数iを0からsize-1未満まで繰り返す
  - ▶ 変数jを0からn未満まで繰り返す
    - ▶ valueのj番目の値とj+1番目の値を比較して、  
j番目の方が大きかったら、値を交換する（関数swapが使える）
  - ▶ nの値を1減らす

# 【課題1-3】

[mainの処理]

//↓配列vが宣言済みの場合は不要

```
int v[10] = {9, 7, 6, 0, 1, 5, 2, 3, 4, 8};
```

```
show(v, 10);
```

```
bubble(v, 10);
```

```
show(v, 10);
```

[実行結果]

```
9 7 6 0 1 5 2 3 4 8          (←ソート前の出力)
```

```
0 1 2 3 4 5 6 7 8 9          (←ソート後の出力)
```

まだ余裕のある人は…

# 【課題1-4】

ソートする配列を以下のように変更して、バブルソートの動作確認をしてみましょう。

- ▶ 配列vの初期値を変えてみる
- ▶ 配列vの要素数を20など増やしてみる