Lesson 4
Sorting

LECTURER: HARVARD TSENG

Look Back

Let’s look back to lesson 3. How we sort(排序) were not a clever way.

We need new solution.

Sorting Algorithms 排序法

Bubble sort(氣泡排序)

Selection sort(選擇排序)

Insertion sort(插入排序)

Merge sort(合併排序)

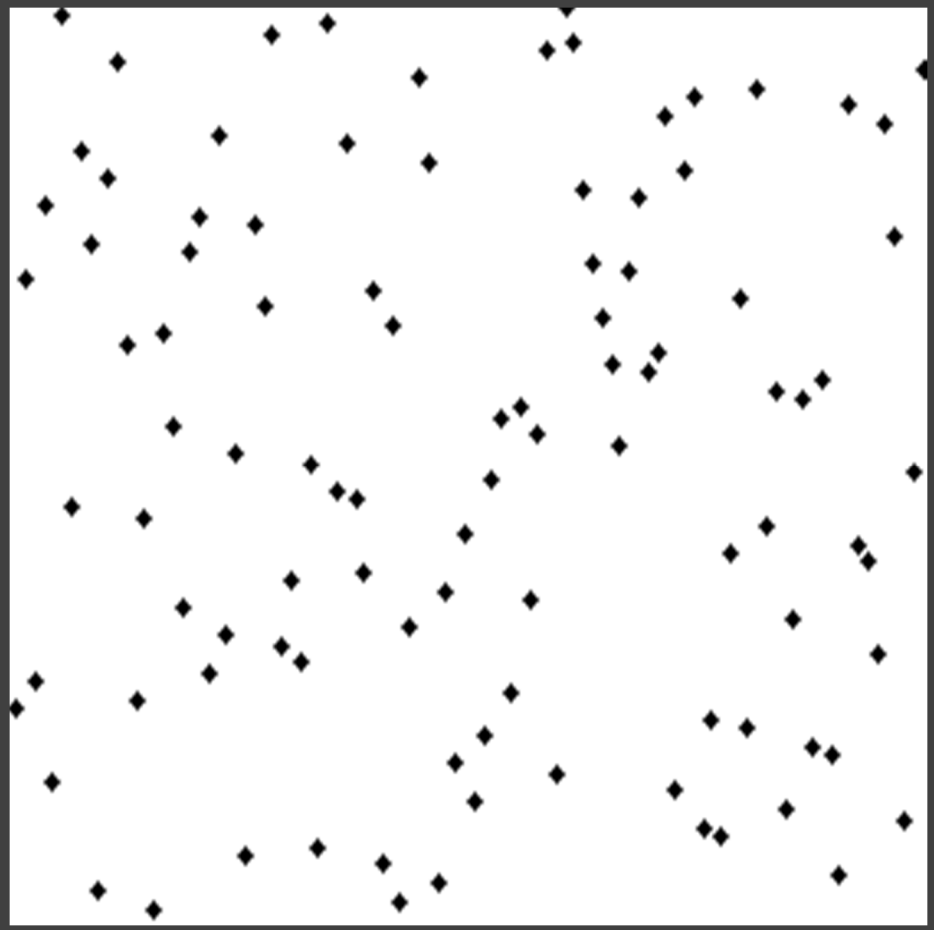
Quick sort(快速排序)

……

Bubble Sort

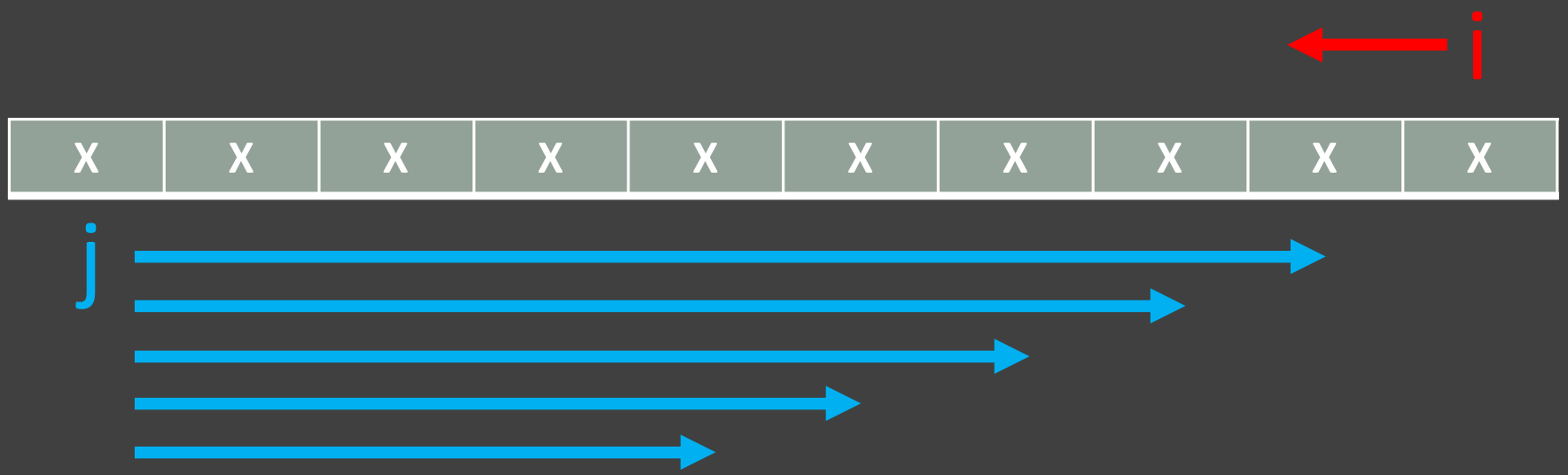
If present number is bigger than next number, then swap(交換).

Bubble Sort



Source: Wiki

Bubble Sort



How to swap?

```
int temp = a;

a = b;

b = temp;
```

Array 陣列

Can carry several same type values together. More convenient to manage values.

How to define array?

Just like define a variable.

float a = 12.34;

int nums[10] = {7, ,9};

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
7	X	9	X	X	X	X	X	X	X

How to use array?

```
int nums[5] = {7, 9, -1};
int temp = nums[1];        //temp = 9
nums[4] = 10;        //nums[] = {7, 9, -1, x ,10}
cout<<nums[0];
```

Exercise 1

Use bubble sort to sort global array small to big.

```
int nums[10] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
int main(){
    //bubble sort
}
```

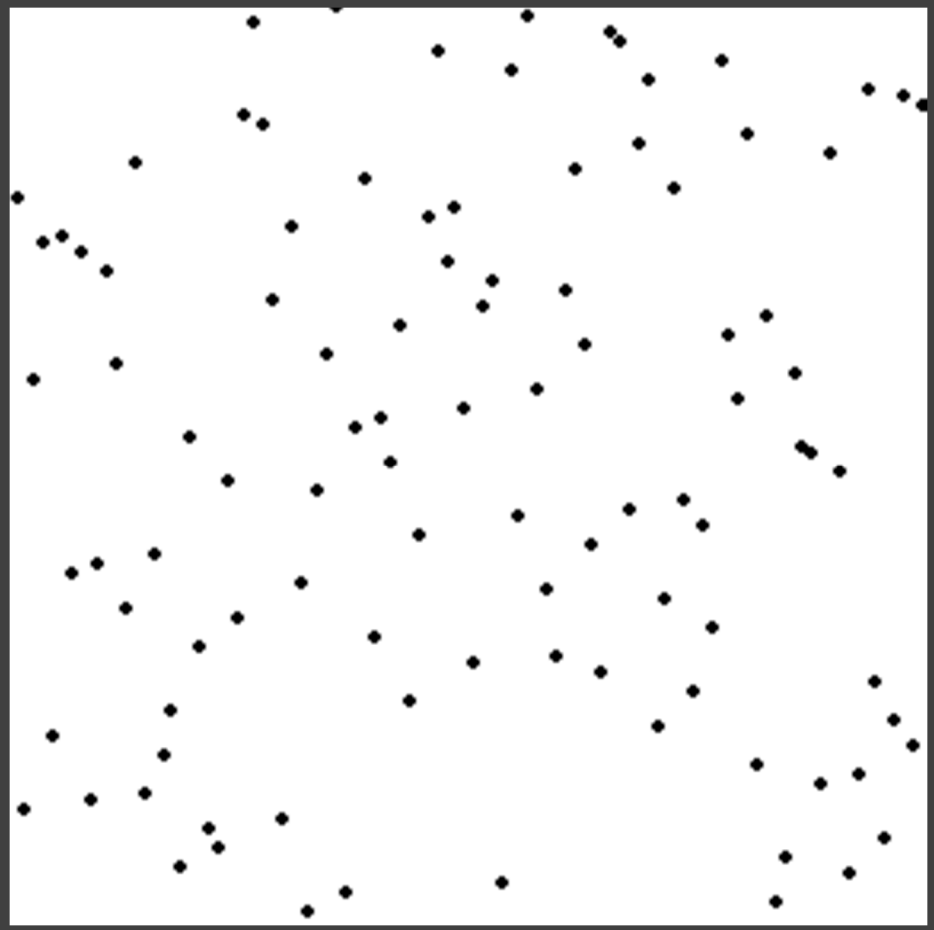
<Tips>

Use function & for loop to simplify your code.

Selection Sort

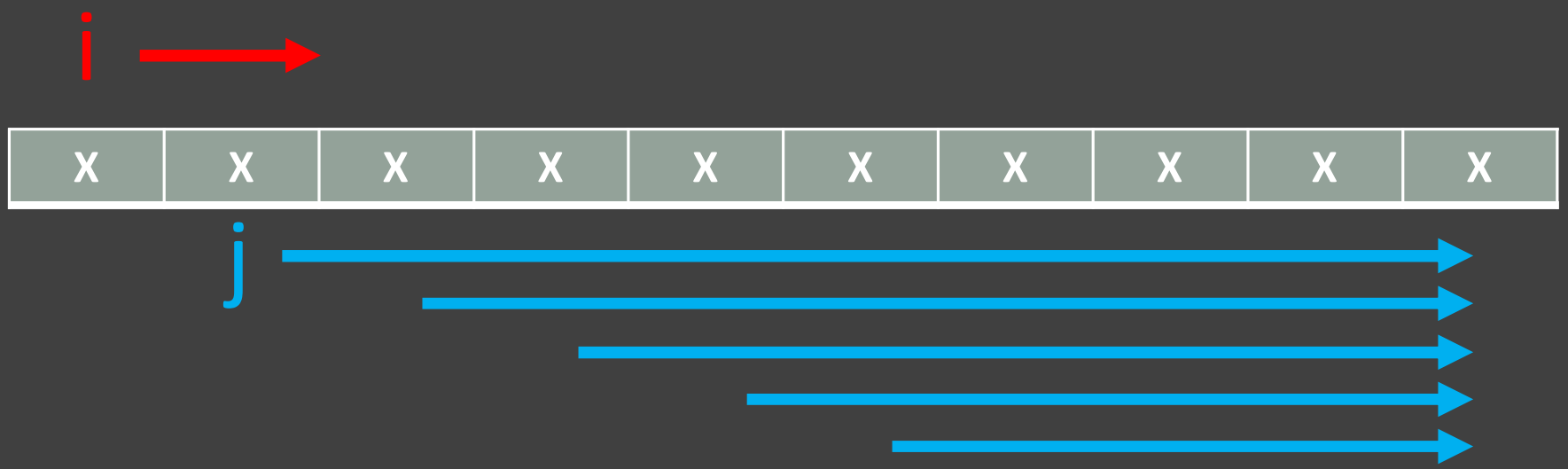
Find the smallest number and put to first place. Second smallest number put to second place, and so on.

Selection Sort



Source: Wiki

Selection Sort



Exercise 2

Use selection sort to sort global array small to big.

```
int nums[10] = {1, 2, 3, 4, 5, 4, 3, 2, 1, 0};
int main(){
    //selection sort
}
```

<Tips>

You need a variable to remember the location of smallest value.