Lec 9
10/30 ~ 11/06

Sorting

① Sort → ascending → ② bubble, ③ selection, ④ insertion, ⑤ merge, ⑥ quick, ⑦ count-sort, ⑧ radix, ⑨ heap

① bubble ② selection

⑩ 1 million sort

bubble

(Name)

⑤ 4 3 2 1	⑥ 4 3 2 1	⑦ 2 3 1 4 5
3 2 1 4	3 2 1 5	2 3 1 4 5
3 2 1 4 5	3 2 4 5	2 3 1 4 5
3 2 1 5	3 2 4 5	2 1 3 4 5
3 2 4 5	3 2 1 5	2 1 3 4 5
3 2 1 5		1 2 3 4 5
3 2 1 5		1 2 3 4 5
3 2 1 5		

⑧ 4 3 2 1 5

⑨ 3 2 1 5 4

⑩ 3 2 1 5 4

⑪ 3 2 1 5 4

⑫ 3 2 1 5 4

⑬ 3 2 1 5 4

⑭ 3 2 1 5 4

⑮ 3 2 1 5 4

⑯ 3 2 1 5 4

⑰ 3 2 1 5 4

⑱ 3 2 1 5 4

⑲ 3 2 1 5 4

⑳ 3 2 1 5 4

for n

for i

for j

if (arr[j] > arr[j+1])

temp = arr[j]

arr[j] = arr[j+1]

arr[j+1] = temp

arr.length-1-i

5=4

4=3

3=2

2=1

1=5

```
public static void Sort(int[] arr) {
    // TODO Auto-generated method stub
    for (int n = 0; n < arr.length; n++) {
        for (int i = 0; i < arr.length-1-i; i++) {
            if (arr[i] > arr[i+1]) {
                int temp = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = temp;
            }
        }
    }
}
```

i=0 ① 4 3 2 5 1

i=1 ② 3 4 2 5 1

i=2 ③ 3 2 4 5 1

i=3 ④ 3 2 1 4 5

i=4 ⑤ 3 2 1 5 4

int[] arr = {4, -1, 5, 3, 2, 1, 7};

0 1 2 3 4 5 6

4 -1 5 3 2 1 7

-1 4 5 3 2 1 7 min=1 → 1

-1 1 5 3 2 4 7 min=2 → 2

-1 1 2 5 3 4 7 min=3 → 3

-1 1 2 3 5 4 7 min=4 → 4

-1 1 2 3 4 5 7 min=5 → 5

-1 1 2 3 4 5 7 min=7 → 7

-1 1 2 3 4 5 7

i=0

i=1

i=2

i=3

i=4

i=5

i=6

Find min!

min = i

for(int i = start; i<arr.length; i++)

if (arr[i] < arr[min])

min = i

1 2 3 5 8 9

1 2 3 5 8 9

1 2 3 5 8 9

1 2 3 8 9

1 2 3 8 9

i=1 → 2

i=2 → 3

i=3 → 5

i=4 → 8

i=5 → 9

i=6 → 8

i=7 → 9

i=8 → 8

i=9 → 9

i=10 → 8

i=11 → 9

i=12 → 8

i=13 → 9

i=14 → 8

i=15 → 9