

Comparison Sorting: Executive Summary

Sandra Batista, Mark Redekopp, and David Kempe

1.1–1.2

Sorting Stability

- A sort is **stable** if the order of equal items in the original list is maintained in the sorted list
 - Good for searching with multiple criteria

List	7,a	3,b	5,e	8,c	5,d
index	0	1	2	3	4

Original

List	3,b	5,e	5,d	7,a	8,c
index	0	1	2	3	4

Stable Sorting

List	3,b	5,d	5,e	7,a	8,c
index	0	1	2	3	4

Unstable Sorting

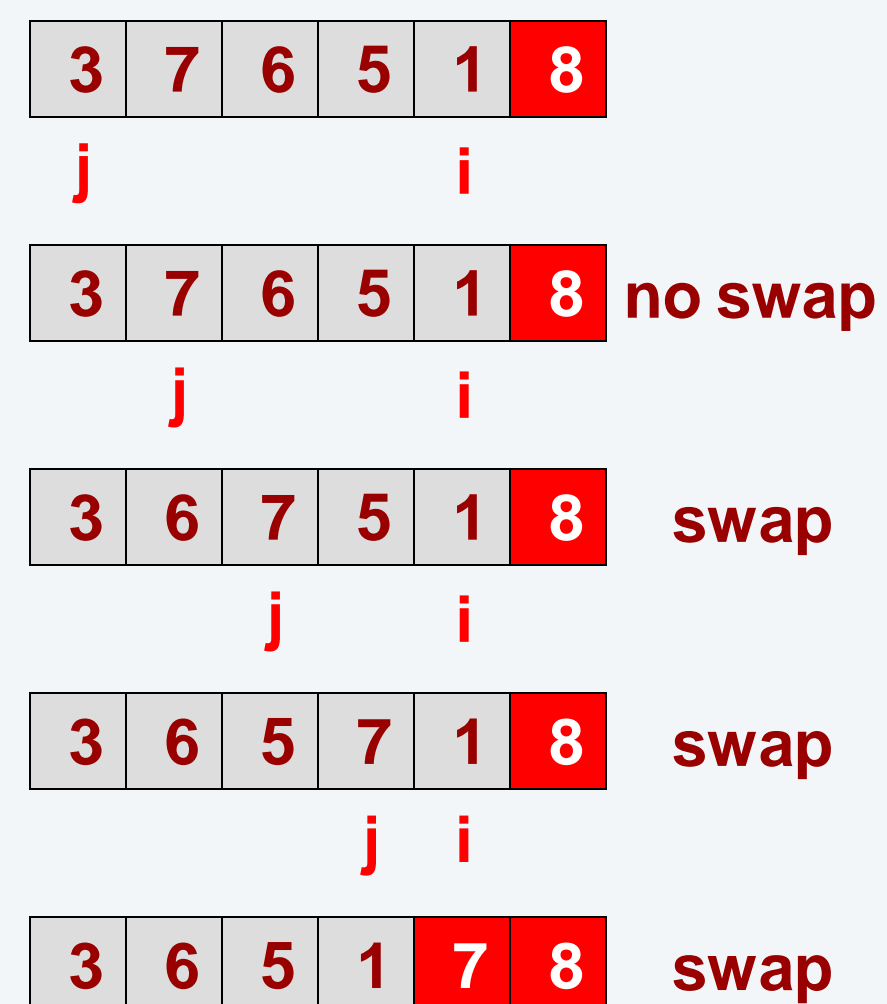
Bubble Sort Algorithm

```
void bsort(vector<int> mylist)
{
    int i ;
    for(i=mylist.size()-1; i > 0; i--){
        for(j=0; j < i; j++){
            if(mylist[j] > mylist[j+1]) {
                swap(j, j+1)
            }
        }
    }
}
```

Pass 1

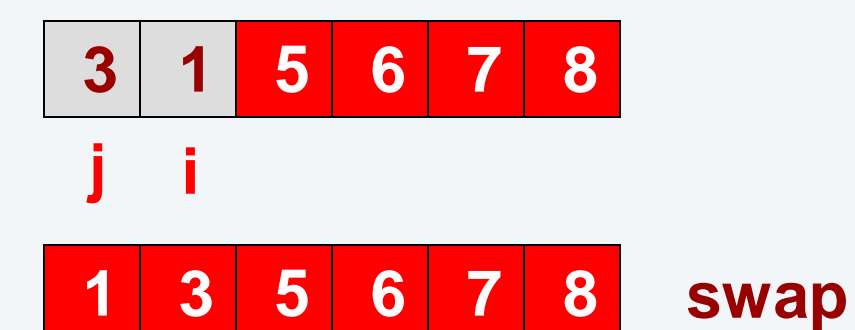


Pass 2



...

Pass n-2



Selection Sort Algorithm

```
void ssort(vector<int> mylist)
{
    for(i=0; i < mylist.size()-1; i++){
        int min = i;
        for(j=i+1; j < mylist.size; j++){
            if(mylist[j] < mylist[min]) {
                min = j
            }
        }
        swap(mylist[i], mylist[min])
    }
}
```

Pass 1 min=0

7	3	8	6	5	1
---	---	---	---	---	---

i j

Pass 2 min=1

7	3	8	6	5	1
---	---	---	---	---	---

i j

Pass 3 min=1

7	3	8	6	5	1
---	---	---	---	---	---

i j

Pass 4 min=1

7	3	8	6	5	1
---	---	---	---	---	---

i j

Pass 5 min=5

7	3	8	6	5	1
---	---	---	---	---	---

i j

Pass 6 swap

1	3	8	6	5	7
---	---	---	---	---	---

Pass 2 min=1

1	3	8	6	5	7
---	---	---	---	---	---

i j

min=1

1	3	8	6	5	7
---	---	---	---	---	---

i j

min=1

1	3	8	6	5	7
---	---	---	---	---	---

i j

min=1

1	3	8	6	5	7
---	---	---	---	---	---

i j

min=1

1	3	8	6	5	7
---	---	---	---	---	---

i j

swap

Pass n-2 min=4

1	3	5	6	7	8
---	---	---	---	---	---

 i j

Insertion Sort Algorithm

```
void isort(vector<int> mylist)
{  for(int i=1; i < mylist.size(); i++){
    int val = mylist[i];
    hole = i
    while(hole > 0 && val < mylist[hole-1]){
        mylist[hole] = mylist[hole-1];
        hole--;
    }
    mylist[hole] = val;}
}}
```

