Sorting with Array

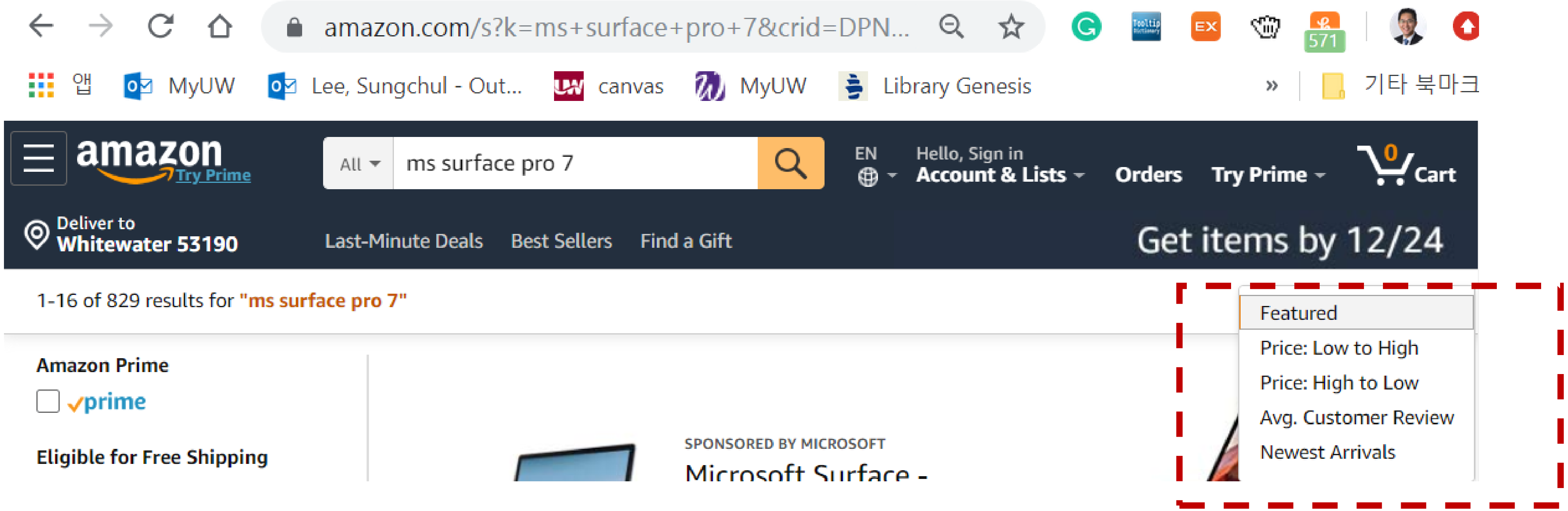
Sungchul Lee

Learning Object

- What is sorting?
- Sorting High to LOW
- Sorting Low to HIGH

Sorting

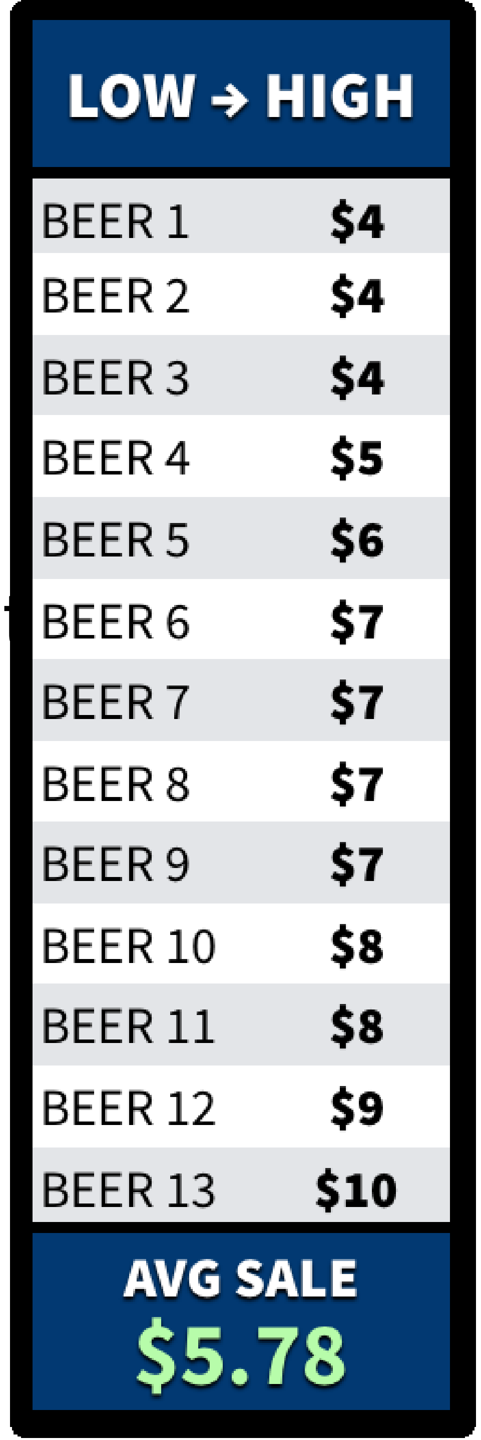
- ➢Sorting is any process of arranging items systematically.
  - ❑**Ordering : sequence ordered**
  - ❑Categorizing : grouping



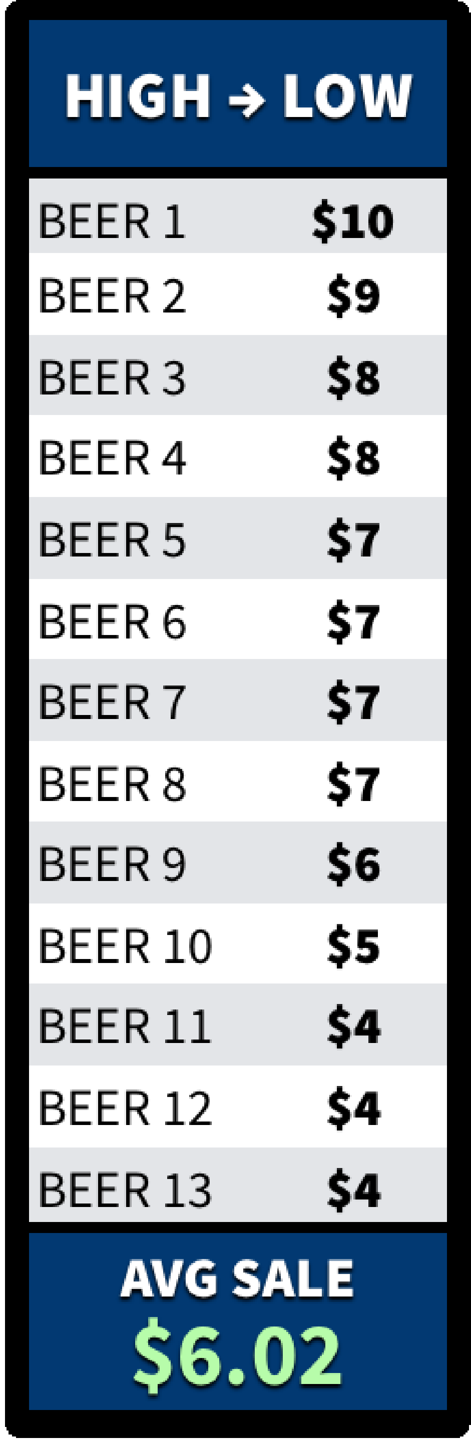
1-16 of 829 results for "ms surface pro 7"

sorting

- ➢Low to Hight
  - ❑Making ascending order
  - ❑Arranging elements from the smallest to the largest number
- ➢Hight to Low
  - ❑Making descending order
  - ❑Arranging elements from the largest to the smallest number



LOW → HIGH	
BEER 1	$4
BEER 2	$4
BEER 3	$4
BEER 4	$5
BEER 5	$6
BEER 6	$7
BEER 7	$7
BEER 8	$7
BEER 9	$7
BEER 10	$8
BEER 11	$8
BEER 12	$9
BEER 13	$10
AVG SALE $5.78	



HIGH → LOW	
BEER 1	$10
BEER 2	$9
BEER 3	$8
BEER 4	$8
BEER 5	$7
BEER 6	$7
BEER 7	$7
BEER 8	$7
BEER 9	$6
BEER 10	$5
BEER 11	$4
BEER 12	$4
BEER 13	$4
AVG SALE $6.02	

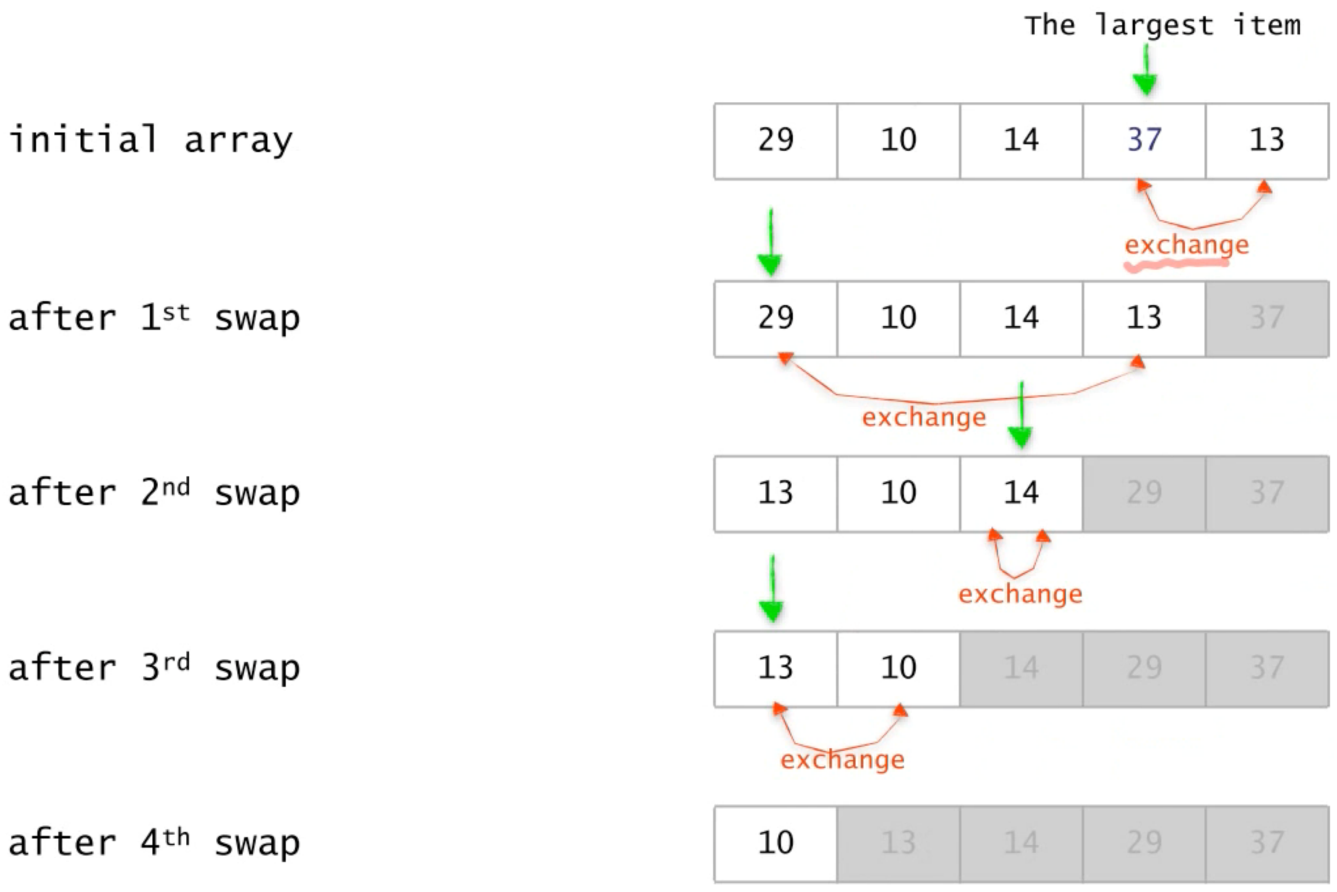
Selection Sorts

- The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

  1) The subarray which is already sorted.
  2) Remaining subarray which is unsorted.

Selection Sorts Algorithm

- Loop
    - Find the index which has a max value
    - **Swap** the value to rightmost elements
    - Exclude rightmost element



The largest item

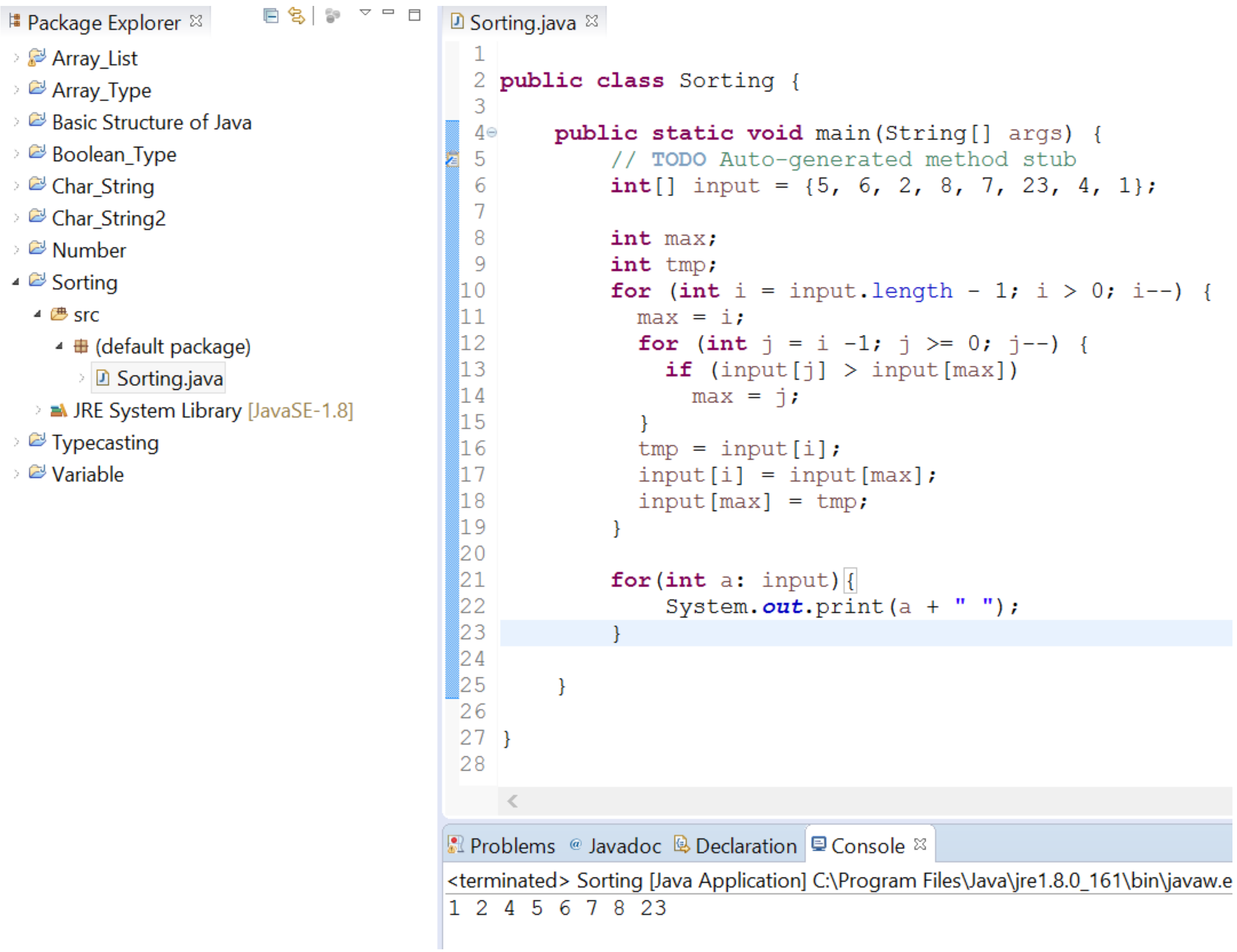
initial array	29	10	14	37	13
after 1st swap	29	10	14	13	37
after 2nd swap	13	10	14	29	37
after 3rd swap	13	10	14	29	37
after 4th swap	10	13	14	29	37

Practice

1. Make a new project
   - ❑Project name: **Sorting**
   - ❑Create a new Class File
     - ❖Class name: **Sorting**
   - ❑Coding:

```
public class Sorting {
public static void main(String[] args) {
// TODO Auto-generated method stub
    int[] input = {5, 6, 2, 8, 7, 23, 4, 1};
    int max;
    int tmp;
    for (int i = input.length - 1; i > 0; i--) {
      max = i;
      for (int j = i -1; j >= 0; j--) {
                if (input[j] > input[max])
                 { max = j;}
      }
      tmp = input[i];
      input[i] = input[max];
      input[max] = tmp;
    }
    for(int a: input){
                System.out.print(a + " ");
    }
}
```

Practice – Code and Result

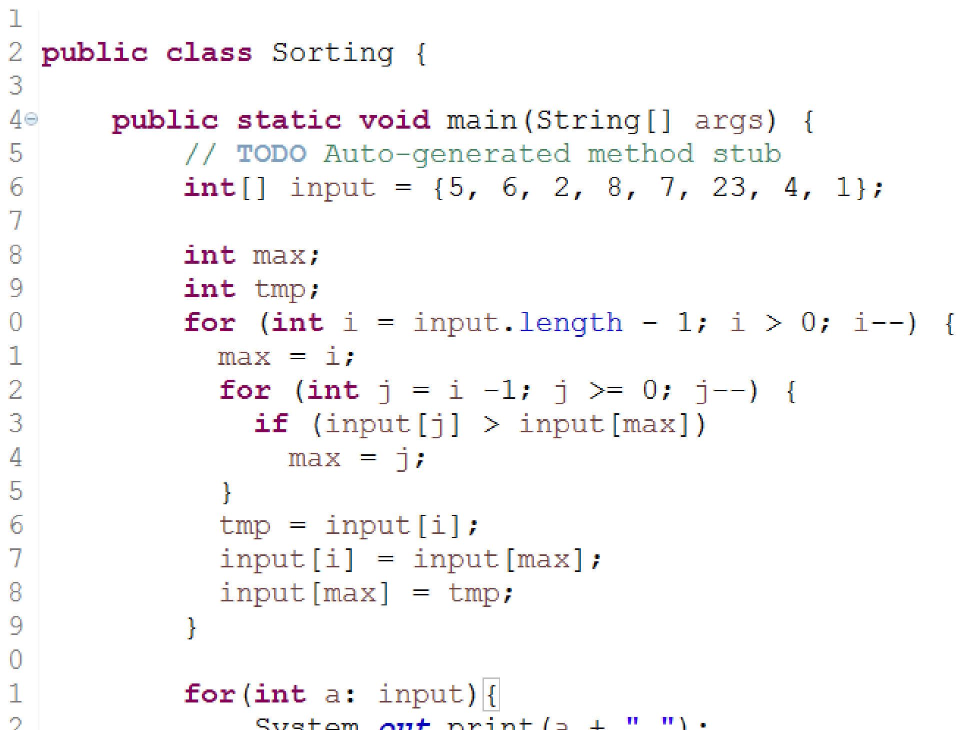


```

public class Sorting {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] input = {5, 6, 2, 8, 7, 23, 4, 1};

        int max;
        int tmp;
        for (int i = input.length - 1; i > 0; i--) {
          max = i;
          for (int j = i -1; j >= 0; j--) {
            if (input[j] > input[max])
              max = j;
          }
          tmp = input[i];
          input[i] = input[max];
          input[max] = tmp;
        }

        for(int a: input){
            System.out.print(a + " ");
        }

    }

}

```

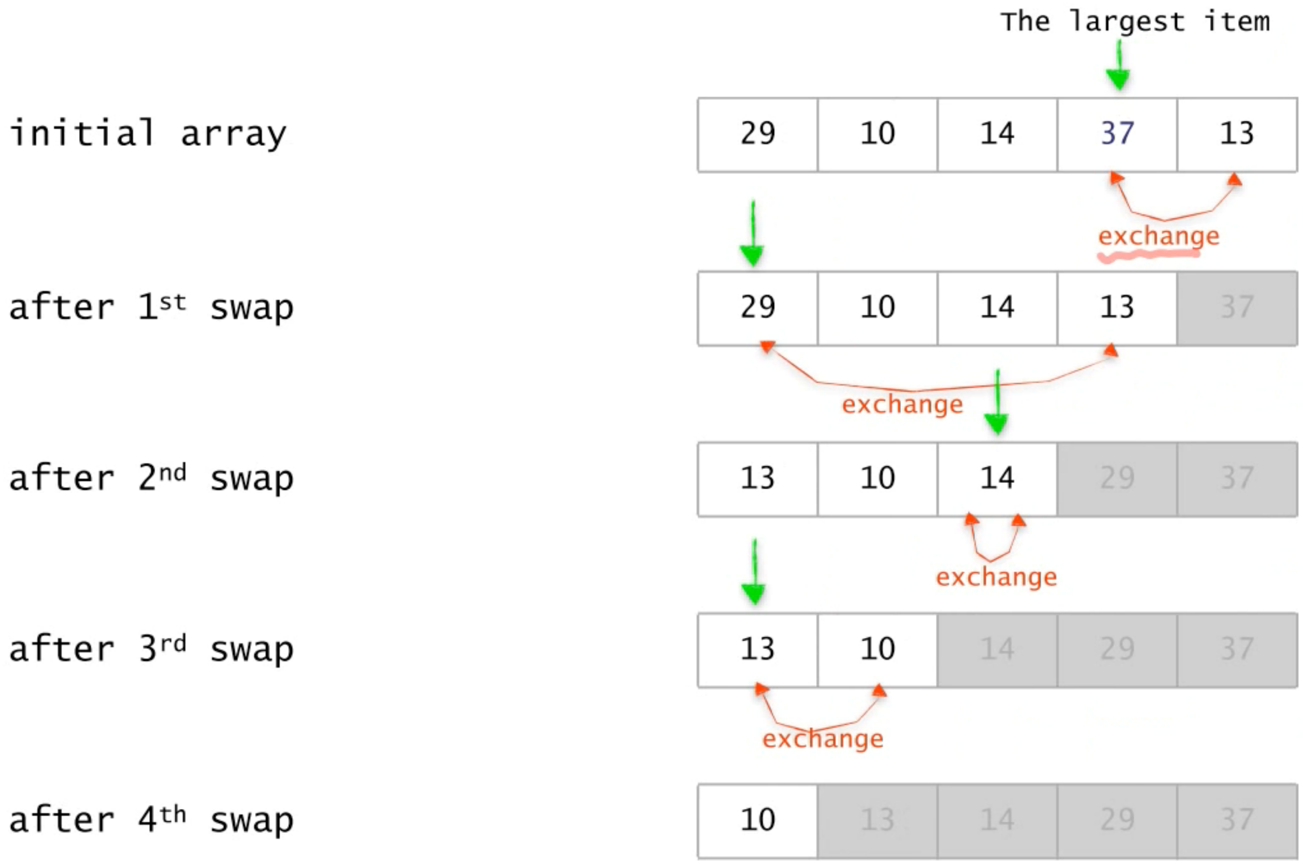
```
<terminated> Sorting [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.e
1 2 4 5 6 7 8 23
```

Summary

- ➢Sorting is any process of arranging items systematically.
  - ❑**Ordering**
    - ❖**sequence ordered**



```
public class Sorting {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] input = {5, 6, 2, 8, 7, 23, 4, 1};

        int max;
        int tmp;
        for (int i = input.length - 1; i > 0; i--) {
          max = i;
          for (int j = i -1; j >= 0; j--) {
            if (input[j] > input[max])
              max = j;
          }
          tmp = input[i];
          input[i] = input[max];
          input[max] = tmp;
        }

        for(int a: input){
          System.out.print(a + " ");
```



The largest item

initial array: 29 | 10 | 14 | 37 | 13

exchange

after 1st swap: 29 | 10 | 14 | 13 | 37

exchange

after 2nd swap: 13 | 10 | 14 | 29 | 37

exchange

after 3rd swap: 13 | 10 | 14 | 29 | 37

exchange

after 4th swap: 10 | 13 | 14 | 29 | 37