Selection Sort

→ Take largest element and swap it with it's correct index.

Example

0 1 2 3 4
4, 5, 1, 2, 3
4, 3, 1, 2, 5
2, 3, 1, 4, 5
2, 1, 3, 4, 5
1, 2, 3, 4, 5 → Sorted.

⇒ Complexity

# of comparison.

$$0 + 1 + 2 + \dots + (n-1)$$

$$\frac{(n-1) * (n-1+1)}{2}$$

$$= \frac{n(n-1)}{2} = \frac{n^2 - n}{2} = \boxed{O(n^2)}$$

best and worst case.

* Performs well on small list

* Pseudo code

```
for (int i=0; i < arr.length; i++)
        int last = arr.length-i-1;
        int max = getmaxInt( arr, 0, last)
        Swap(arr, maxIndex, last)
```

```
int getmax(arr, s, e)
    int max = s;
    for (i=s; i<=e; i++)
        if (arr[max] < arr[i])
            max = i
    return max;
```

```
Swap( arr, first second )
    temp = arr[first]
    arr [first] = arr [second]
        arr [second] = temp
```