//Thi Thanh Tra Kieu #261066512

2. Quick Sort

Input: array not sorted type int, startIndex type int, endIndex type int
Output: array sorted

```
function quickSort(array, startIndex, endIndex){
    if(startIndex >= endIndex                                          => O(1)
        return;
    declare pivotIndex = partition (array, startIndex, endIndex);      => O(nlogn)
    quickSort(arr, startIndex, pivotIndex-1);                          => O(nlogn)
    quickSort(arr, pivotIndex+1, endIndex);                            => O(nlogn)
}
```

T(n) = O(1) + O(nlogn) + O(nlogn) + O(nlogn) = O(nlogn)

Input: array not sorted type int, startIndex type int, endIndex type int
Output: index type int (pivot in the next round)

```
function partition(array, startIndex, endIndex){
    choose 1 random index in array;
    swap array at random index and array at endIndex;
    declare and initialize pivotValue = array at endIndex;
    declare and initialize i = 0, j = endIndex-1;
    while(i<j){
        if(array at i <= pivoteValue)
            increment i by 1;
        else
            swap array at i and array at j;
            decrement j by 1;
    }
    if(array at i > pivotValue)
        swap array at i and array at endIndex;
    else
        increment i by 1;
        swap array at i and array at endIndex;
    return i;
}
```