Quick Sort

Friday, October 27, 2023 9:35 AM

EXAMPLE ONE

LeftIndex - Keeps incrementing until it finds value **larger** than pivot
RightIndex - Keep decrementing until it finds value **smaller than** pivot

[3, 12, 15, 9, 6, 2, 10]
[3, 12, 15, 9, 6, 2, 10] <-swap
[3, 2, 15, 9, 6, 12, 10]
[3, 2, 15, 9, 6, 12, 10] <-swap
[3, 2, 6, 9, 15, 12, 10] <- first round of quick sort is complete
[3, 2, 6, 9, 15, 12, 10] <-rightIndex > leftIndex (condition to stop)

[3, 2, 6] and [9, 15, 12, 10] <- partitions

[3, 2, 6]
[3, 2, 6] <-swap
[2, 3, 6]
[2, 3, 6] <-rightIndex > leftIndex (condition to stop)

[9, 15, 12, 10]
[9, 15, 12, 10] <- swap
[9, 10, 12, 15]
[9, 10, 12, 15]<-rightIndex > leftIndex (condition to stop)

EXAMPLE TWO

[15,5,24,8,1,3,16,10] <-swap
[10,5,24,8,1,3,16,15]
[10,5,24,8,1,3,16,15]
[10,5,24,8,1,3,16,15] <- swap
[10,5,3,8,1,24,16,15]
[10,5,3,8,1,24,16,15] <- swap
[10,5,3, 1,8,24,16,15]
[10,5,3 1,8,24,16,15] <-rightIndex > leftIndex (condition to stop)

[10,5,3 1] and [8,24,16,15] <- partitions

[10,5,3 1] <- swap
[1,5,3 10]
[1,5,3 10] <- swap
[1,3,5 10]
[1,3,5 10] <-rightIndex > leftIndex (condition to stop)

[8,24,16,15]
[8,24,16,15] <-swap

[8,15,16,24]
[8,15,16,24] <-rightIndex > leftIndex (condition to stop)

StackAsArray

maxSize

Constructor:
Int* stack = new int[maxSize]
top = -1

```
Pop(){
// checking if stack is not empty
        If(top >= 0)
                top--;
}

Push(int x){
// checking if stack is full
        If(top+1 < maxSize){
                Top++
                Arr[top] = x
        }
}
```