

Stacks and Queues

S – Stack, Q – Queue, SQ – Stack/Queue

Last Updated:10/15/2021

SQ1. Short answers:

a) (3pts) A stack has M items. Fill in the total **time complexity** to do N **push** operations on it.

Θ (.....)

b) The time complexity to insert in an array-based queue with X items is Θ (_____).

c) The space complexity for a pop() operation in a list-based stack with T items is Θ (_____).

S1. Write code for the following stack operations, implementing the array-based stack:

```
Stack newStack(int N);  
void push(Stack * s, int item);  
int pop(Stack * s);
```

Use the given struct:

```
struct stack_array{  
    int * items;  
    int top; //index AFTER last item  
    int capacity;  
};  
typedef struct stack_array Stack
```

S2. Letters are pushed on a stack in order: R A N D O M O P S. Specify where to insert pop operations (shown by ‘*’) among the pushes of the given letters, in order to produce the output: ADONOMSPR .

You can only do this process once. That is, you cannot take the output produced and then pass it again through the stack.

S3. Letters are pushed on a stack in order: R E A S O N . Specify where to insert pop operations (shown by ‘*’) among the pushes of the given letters, in order to produce the output: O S E R A N.

You can only do this process once. That is, you cannot take the output produced and then pass it again through the stack.

S4. Starting with an empty stack, show the stack after each operation: 5,9,3,*,*,10,*,12,*

(Here 5,9,3,*,*,... indicate the operations: push(5), push(9), push(3), pop(), pop(),...)

S5. Leetcode (NOT part of the quiz)

- [20. Valid Parentheses](#)

- [1544. Make The String Great](#)

- [1190. Reverse Substrings Between Each Pair of Parentheses](#)

Q1. Write code for the following queue operations, implementing the array-based queue.

```
bool put(Queue * Q, int val);  
bool get(Queue * Q, int* ret);
```

Use the given struct:

```
struct queue_array {  
    int capacity;  
    int size;  
    int first_index; // index of first item  
    int last_index; // index AFTER last item  
    int * items;  
};  
typedef struct queue_array Queue;
```

Q2. (9 pts) Assume that a FIFO Queue is implemented using an array of size 3. Show the array after each of the operations: 5,9,3,*,*,10,*,12,*

(Here 5,9,3,*,*,... indicate the operations: put(5), put(9), put(3), get(), get(),...)

