

Queue Using Two Stacks

Declaration

I hereby declare that all the work done in this project titled "Queue Using Two Stacks" is of my independent effort.

Chapter 1: Introduction

A queue (FIFO structure) can be implemented by two stacks (LIFO structure) . Assume that each operation of push or pop takes 1 unit of time. My job is to tell the time taken for each dequeue.

Chapter 2: Algorithm Specification

Algorithm:

The task can be sloved in the following way:

1. Start from two empty stacks $s1$ and $s2$.
2. When element e is enqueued, it is actually pushed onto $s1$.
3. When we are supposed to dequeue, $s2$ is checked first. If $s2$ is empty, everything in $s1$ will be transferred to $s2$ by popping from $s1$ and immediately pushing onto $s2$. Then we just pop from $s2$ -- the top element of $s2$ must be the first one to enter $s1$ thus is the first element that was enqueued.

Chapter 3: Testing Results

Sample Input:

```
10
I 20
I 32
O
I 11
O
O
O
I 100
I 66
O
```

Sample Output:

```
20 5
32 1
11 3
ERROR
100 5
```

Time Complexity:

1. Enqueue Operation ('I'):

- The enqueue operation involves pushing an element onto the stack `s1`.
- Time complexity: $O(1)$

2. Dequeue Operation ('O'):

- If stack `s2` is not empty, the dequeue operation involves popping from `s2`.
- If stack `s2` is empty, elements from `s1` are transferred to `s2`, and then the dequeue operation is performed.
- In the worst case, all elements are transferred from `s1` to `s2`.
- Time complexity: $O(N)$

Space Complexity:

1. Stacks (`s1` and `s2`):

- Two stacks of size `N` are used to simulate the queues.
- Space complexity: $O(N)$

2. Other Variables:

- Integer variables (`number`, `time`, `tos1`, `tos2`, `op`, `i`) and arrays are of constant size.
- Space complexity: $O(1)$