Name :- Zubayer Ahmed Zidhan Laskar, Roll :- 20071000 7062

Subject :- Data Structures & Algorithms , Date :- 29/10/2021

Semester :- 3rd

CIE-1

1. Ans :- A stack is a type of data structure in which new elements are added and removed from the same side of the array called "top". A stack exhibits LIFO (Last In first Out) property.

The push algorithm for a stack implemented using an array is as follows :-

Procedure push (S, maxsize, top, element)

1. If (top == maxsize - 1)
2. Print ("Stack Overflow!").
3. Return.
4. End If
5. top := top + 1
6. S[top] := element.
7. Return.

2. Ans :- A queue is a type of linear data structure where all the insertions are made at one end of the list (known as "rear"), and, all the deletions are made at the other end of the list (called "front"). A queue exhibits FIFO (First in First Out) property.

Scanned with CamScanner

The algorithm for delete operation on a circular queue implemented using array is as follows.

Procedure Dequeue (Q, front, rear, maxsize)

1. If (front == -1) Then
2. Write ("Queue Underflow")
3. Return.
4. End If
5. Write ("The element to be deleted is: ", Q[front]);
6. If (front == rear) Then
7. front := -1
8. rear := -1
9. Else
10. front := (front + 1) % maxsize
11. End If
12. Return

3.> Ans:- Asymptotic complexity of an algorithm is the anlysis of the concerned algorithm when the size of its input/output is very high (∞).

Given, 'N' is the max. size of the array

Let, 'n' be the no. of elements present in the array.

~~The algorithm for insertion at a particular posn is as follows~~

~~Procedure Insert (N, n, element, position, arr~~

~~1. If n == N, Then~~

~~2. Return~~

~~3. Else~~

~~4. Repeat for i in n, n-1, n-2, --, position)~~

~~5. arr[i-1] := arr[i]~~

~~6.~~

3.) Ans: [Contd.]

The algorithm for insertion at a particular position is as follows:-

<u>Procedure Insert (N, n, element, position, arr)</u>

1. If $n == N$, Then
2. Return
3. Else
4. Repeat for i in $n, n-1, n-2$ —, position
5. $arr[i] := arr[i-1]$
6. $arr[position] := element$
7. $n{+}{+}$
8. Return.
9. EndIf

frequency count $= 1 + (n - position + 1) + (n - position + 1) + 1 + 1$

$= 2(n - position + 1) + 3 \approx 2n + 3$ [if $n >> pos^n$]

The worst case will be when we have to insert at index 0, in that case total steps

$= 1 + (N+1) + (N+1) + 1 + 1$

$= 2N + 5$, which of the order $O(N)$

∴ Worst case scenario is, $O(N)$

4) Ans:- The minimum no. of stacks required for implementing a queue is two (2).

Let the two stacks be $S1$, S_2.

Initially,

S_1

0	1	2	3	4 — —
10	20	30	40	

top = 3

S_2

0	1	2	3	4 — —

top = 0

Insert 50

S_1

0	1	2	3	4
10	20	30	40	50

top = 4

Now pop from S_1 and push all to S_2

0	1	2	3	4
50	40	30	20	10

top = 4

So, '10' was the first entered element is at the top of S_2, and so in case of deletion '10' will be deleted first, fulfilling the FIFO criteria of a queue.

And, ~~xw~~ new elements will be added to S_1 and elements will be popped from S_2, when S_2 is empty, all elements are popped from S_1 and pushed to S_2. And, thus queue is implemented using two stacks.

—

5.) Ans: We can use as a character (char) stack say, S. And then we can traverse the string expression (input) and for every starting bracket i.e. "(" we push it to the stack. And for every closing bracket ")" we pop from the stack. ~~If the popped element matches with "(" then we continue to the next character in the~~

If on reaching the end of the string, there are still some elements left in the stack, then the string of parentheses is not balanced, else, the string is balanced.

—x—