International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

Comparison of Stack and Queue Data structures

Regi Maliyekkel S[1], Anila M[2]

[1]*HSST Computer Science, GHSS Vattenad, Palakkad, Kerala, India*
[2]*Asst Professor on Conract, University Centre, Manjery, Kerala, India*

Abstract - *This paper describes the comparative study of the very familiar Data structures Stack and Queue. While solving problems using computers, the data may have to be processed. These data may be of atomic type or grouped (aggregate). The grouped data are refereed using arrays and structures. Stack and Queue are non-primitive data structures and are used for storing data elements and are actually based on some real world equivalent. The main differences between stack and queue are that stack uses LIFO (last in first out) method to access and add data elements whereas Queue uses FIFO (First in first out) method to access and add. Based on the analysis, a comparative study is being made thus the user can easily understand the working principle and operations of these two data structures.*

***Key Words*: Stack, Push, Pop, Queue, Deletion, Insertion, Comparison**

1. INTRODUCTION

The concept of data structure is similar to the collections of plates in a stand, a set of disks in a CD pack and a queue in which a new person can join the queue only at the rear end. In computer science Data structure is a particular way of organizing similar or dissimilar data items logically and can be considered as a single unit. Since data structures represents collections of data, these are closely related to computer memory, as it is the storage space for data. Here give a comparison study on two very familiar data structures Stack and Queue.

1.1 Stack

The Stack is formed by placing each item one over the other. We can say that the items are added at the top position. And also we can remove that item which is placed at last. This is the Last-In-First-Out (LIFO) principle. The Stack follows LIFO principle. In Stack all Insertions and deletions of data items are done at one end called Top.

Implementation of stack using array

Stack can be implemented as array, in such a case, there is a limit to the number of elements that can be represented by the stack and it depends on the size of the array.

STACK

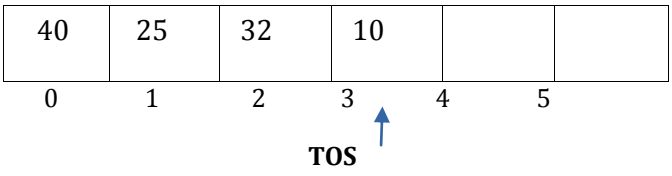


Figure -1

In the above figure- 1 TOS indicates Top Of Stack (Last position of the element in the stack). So by STACK[TOS] indicates the value of the last data item of the stack. Initially the value of TOS set to -1.

1.1.1. Operations on Stack

Two Operations are possible on stack Push and Pop. Push is for inserting data items into stack and Pop is for removing data items from it.

Push Operation

Push Operation is the process of inserting a new data item into Stack. The following Algorithm shows the Push Operation on a Stack. For this consider an array Stack[S] that implements a stack, where S is the maximum size of the array.

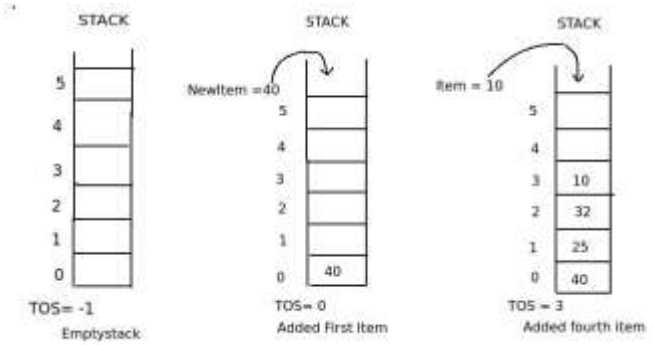


Figure - 2

Step 1: If TOS< S Then (For checking Space availability)

Step 2: TOS=TOS+1

Step 3: Stack[TOS]= Val

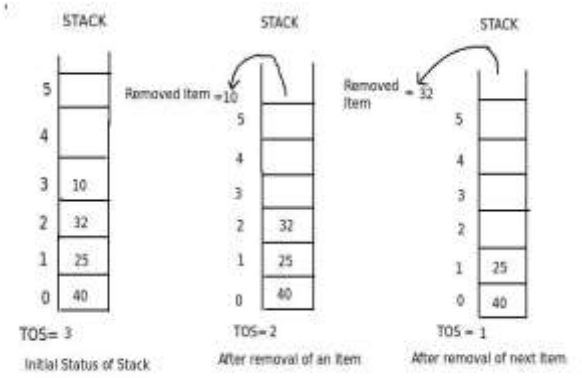
Step 4: Else

Step 5: Print "Stack Overflow"

Step 6: End of If

Pop Operation

Pop operation is the process of deleting an element from the top of Stack. After every deletion, the TOS is decremented by One. The following Algorithm shows the Pop Operation on a Stack. For this consider an array Stack[S] that implements a stack, where S is the maximum size of the array.



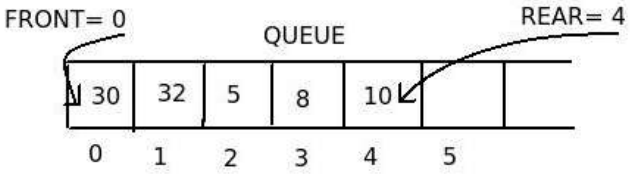


Figure -3

Step 1: If TOS> -1 Then (For checking Empty Stack)

Step 2: Val=Stack[TOS]

Step 3: TOS=TOS-1

Step 4: Else

Step 5: Print "Stack Underflow"

Step 6: End of If

1.2 Queue

We might have became part of queue in our life like billing payment in shops in which one person at the front position billing the payment fist and a new person can join the queue at the rear position. This style of Organizing a group is called Fist-In-First-Out(FIFO) principle. A data structure that follows FIFO is called Queue.

Implementation of Queue using array

Figure shows a queue using array of integers which can accommodate maximum 6 elements. It contains 5 elements and Front value is 0 and Rear value is 4. The first element is represented by QUEUE[FRONT] and the last element is QUEUE[REAR].

1.2.1. Operations on Queue

Insertion and Deletion are the two operations performed on Queue

Insertion Operation

Adding a data element into a queue at the rear end is called Insertion. Following shows Algorithm for Insertion Operations. For this Consider an array QUEUE[S] that implements a queue of Size S. The variable Front and Rear keep track of the position value of Front and Rear elements of the queue.

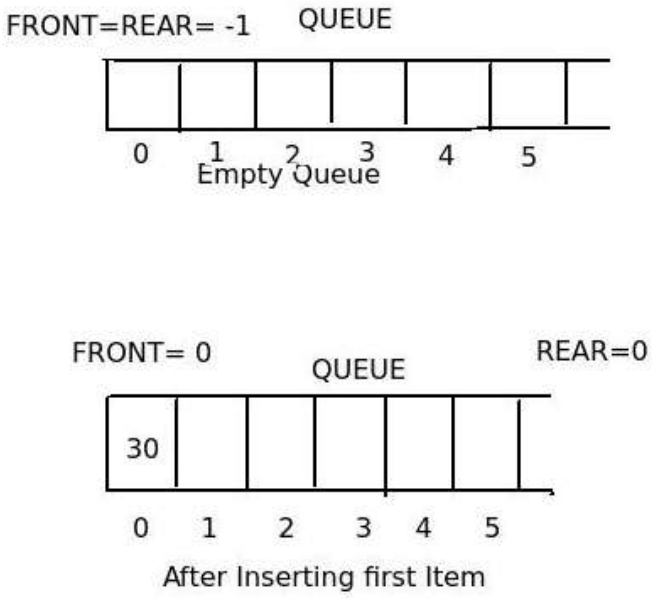


Figure -4

Step 1: If (REAR== -1) Then (For checking Empty Queue)

Step 2: FRONT=REAR=0

Step 3: QUEUE[REAR]=Val

Step 4: Else If(REAR < S) Then

Step 5: REAR=REAR+1

Step 6: QUEUE[REAR]=Val

Step 7: Else

Step 8: Print "Queue Overflow"

Step 9: End of if

Deletion Operation

Removal of data items from the front end of queue is called Deletion operation. Following shows Algorithm for Deletion Operation. For this Consider an array QUEUE[S] that implements a queue of Size S. The variable Front and Rear keep track of the position value of Front and Rear elements of the queue.

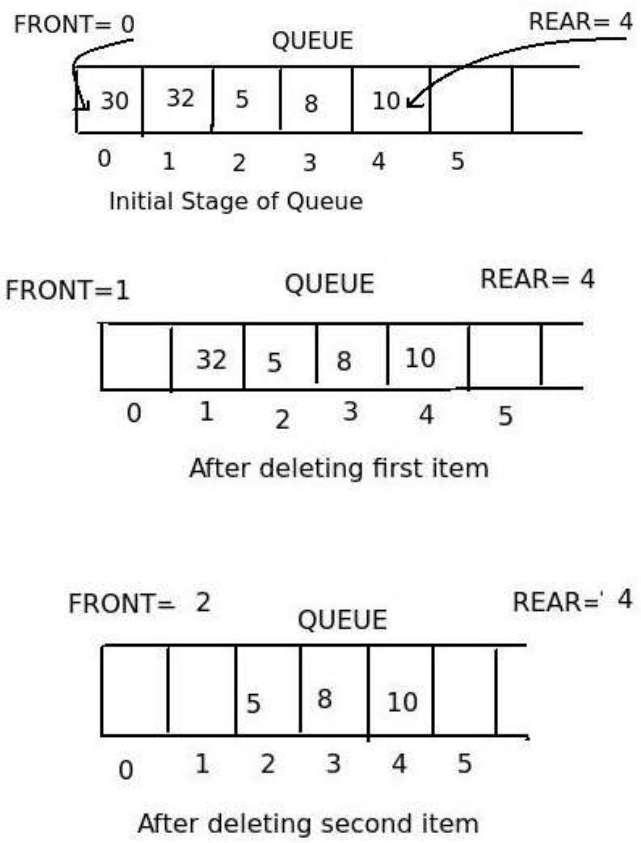


Figure -5

Step 1: If (FRONT. -1 AND FRONT < REAR) Then

(For checking Empty Queue)

Step 2 : Val=QUEUE[FRONT]

Step 3 : FRONT=FRONT+1

Step 4 : Else

Step 5 : Print "Queue Underflow"

Step 6 : End of If

Step 7: If (FRONT>REAR) Then (last element deletion checking)

Step 8: FRONT=REAR= -1

Step 9 : End of if

2. Comparison between Stack and Queue

	Stack	**Queue**
Principle	LIFO	FIFO
Variables	TOS	FRONT REAR
Operations	PUSH POP	Insertion Deletion
Time Complexity	O(1)(Inserting Item) O(1)(Removing Item)	O(1)(Inserting Item) O(n)(Removing Item in worst case)

Table -1

3. CONCLUSION

It can be concluded that we have seen the comparison study of the two popular Data structure Stack and Queue. And also explained the different operations performed on these with proper Algorithms. Both are linear data structures differ in certain ways like working mechanism, structure, implementation, variants but both are used for storing the elements in the list and performing operations on the list like addition and deletion of the elements. Parsing in a compiler. Stack used in Java virtual machine, Undo in a word processor, Back button in a Web browser etc and Queue used in Data Buffers, Asynchronous data transfer (fileIO, pipes, sockets), Traffic analysis etc.

REFERENCES

[1] https://techdifferences.com/difference-between- stack-and-queue.html

[2] https://www.geeksforgeeks.org/difference- between-

stack-and-queue-data-structures/

[3] https://ieeexplore.ieee.org/document/979991 Journals & Magazines Volume: 28 Issue: 1

[4] IEEE Transactions on Software Engineering (Volume: 28 , Issue: 1 , Jan 2002)

[5] Donald Knuth. The Art of Computer Programming Volume1:FundamentalAlgorithms, Addison Wesley, 1997.

[6] RESEARCH PAPER ON STACK AND QUEUE Nitesh, Manbir Singh, Rahul Yadav 2014 IJIRT Volume 1 Issue 7 | ISSN: 2349