

Assignment-1

CS1102

PREPARED BY

YESH RAJAWAT

2019 BTECHCSE 065

Data Structures



Department of engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

2019-2023

September 2020

Question 1: Write a recursive function that takes as input a queue and rearranges it so that it is in reverse order. Hint: deque() the first element, recursively reverse the queue, and the enqueue the first element.

Answer:-

```
import java.util.*;
class Question_1 {

    int front=-1;
    int rear = -1;
    int [] queue ;
    int size;
    void set_size(int size)
    {
        this.size = size;
        queue = new int[size];
    }

    void enqueue(int val)
    {
        if(front == -1 && rear == -1)
        {
            front = 0;
            rear=0;
            queue[rear] = val;
        }
        else if(rear == size-1)
        {
            System.out.println("Overflow");
        }
        else
        {
            rear++;
            queue[rear] = val;
        }
    }

    void deque()
    {
        if(front== -1&& rear== -1)
        {
            System.out.print("Underflow");
        }
    }
}
```

```

        else if(front >=rear)
        {
            front = -1 ;
            rear = -1;
        }

        else
        {
            front++;
        }
    }

void traversal()
{
    if(front==-1 && rear==-1)
    {
        System.out.print("Empty queue");
    }
    else
    {
        for(int i=front;i<=rear;i++)
        {
            System.out.print(queue[i]+" ");
        }
    }
    System.out.println();
}

```

```

void reversal(int []queue)
{
    if(rear==-1 && front==-1)
    {
        return;
    }

    int temp = queue [front];
    deque();

    reversal(queue);
    enqueue(temp);
}

```

```

    }

    public static void main(String... args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the size of queue");
        Question_1 queue = new Question_1();
        int size = scan.nextInt();
        queue.set_size(size);

        for(int i=0;i<size;i++)
        {
            queue.enqueue(scan.nextInt());
        }
        queue.traversal();
        queue.reversal(queue.queue);
        queue.traversal();

        scan.close();}
}

```

Question 2: Build a JAVA program from scratch that reverses the words in a sentence.

Answer:-

In this code I have define three methods excluding push() and pop() and all of the three perform reversing the string but I have used only the method which uses stack implementation

```

import java.util.*;

public class Question_2 {

    void reverse_Sentence_Brute(String s)
    {

```

```

String reverse= "" ;

for(String temp : s.split(" ") )
{
    for(int i=temp.length()-1;i>-1;i--)
    {
        reverse = reverse + temp.charAt(i);
    }
    reverse = reverse + " ";
}
System.out.println(reverse);
}

```

```

void reverse_Sentence_stack(String s)
{
    Stack <String>stack = new Stack<String>();

    String[] a = s.split(" ");

    for(int i=0;i<a.length;i++)
    {
        stack.push(a[i]);
    }
    String reverse = "";
    for(int i=0;i<a.length;i++)
    {
        reverse = reverse + stack.pop() + " ";
    }

    System.out.println (reverse);

}

int top = -1;
String [] stack ;
int max ;

```

```

public Question_2(int n )
{
    max = n;
    stack = new String[n];
}

void push(String enter)
{
    if(top==-1)
    {
        top = 0;
        stack[top] = enter;
    }
    else if(top == max-1)
    {
        System.out.print("Overflow!!!!!!");
    }
    else
    {
        top ++;
        stack[top] = enter;
    }
}

String pop()
{
    String element;
    if(top==-1)
    {
        System.out.println("Underflow");
        return "";
    }
    else
    {
        element = stack[top];
        top --;
        return element;
    }
}

String reverse(String s )
{
    for(String temp : s.split(" "))
    {
        push(temp);
    }
}

```

```

    }
    String reverse_string = "";

    for(int i=s.split(" ").length-1;i>-1;i--)
    {
        reverse_string= reverse_string +pop() + " ";
    }
    return reverse_string;
}

```

```

public static void main(String... args)
{
    Scanner scan = new Scanner(System.in);

    System.out.println("Enter the string you want to enter: ");
    String sentence = scan.nextLine();

    Question_2 reverse_sentence = new Question_2(sentence.length());
    System.out.println(reverse_sentence.reverse(sentence));

    scan.close();}
}

```

Question 3: Use your stack implementation to implement a queue using the Stack.

Answer:-

```
import java.util.Scanner;
```

```

public class Question_3 {

    int size ;
    int [] stack;
    int top=-1;

    void set_Size(int size)
    {
        this.size = size;
        stack = new int[size];
    }
}

```

```

}

void push(int data)
{
    if(top== -1)
    {
        top =0;
        stack[top] = data;

    }
    else if(top == size-1)
    {
        System.out.println("Overflow");
    }
    else
    {
        top++;
        stack[top] = data;
    }
}

int pop()
{
    int element=0;
    if(top== -1)
    {
        System.out.println("Underflow");
        return 0;
    }
    else
    {
        element = stack[top];
        top--;
        return element;
    }
}

void enqueue(int data)
{
    push(data);
}

int deque()
{
    Question_3 stack_1 = new Question_3();
    stack_1.set_Size(top+1);
    for(int i=top;i>-1;i--)

```



```

        {
            stack_1.push(pop());
        }

        int element = stack_1.pop();
        for(int i=stack_1.top;i>-1;i--)
        {
            push(stack_1.pop());
        }
        return element;
    }
    void traversal()
    {
        if(top== -1)
        {
            System.out.println("No element is present");
        }
        else
        {
            for(int i=0;i<=top;i++)
            {
                System.out.print(stack[i]+" ");
            }
        }
    }

    void peek()
    {
        if(top== -1)
        {
            System.out.println("There is no element");
        }
        else
        {
            System.out.println("Peek element is : " +stack[top] );
        }
    }

    void size()
    {
        System.out.println(top+1);
    }

```

```

public static void main(String ... args)

{

    Scanner scan = new Scanner(System.in);
    System.out.println("=====Queue
Test=====\\n");
    System.out.print("Please enter Size of Integer queue: ");
    int capacity = scan.nextInt();
    Question_3 queue = new Question_3();
    queue.set_Size(capacity);

    char ans;
    do{
        System.out.println("=====Queue
Operations=====");
        System.out.println(">> 1. Enqueue");
        System.out.println(">> 2. Dequeue");
        System.out.println(">> 3. Find the peek element in queue ");
        System.out.println(">> 4. Find the number of elements in queue");
        System.out.print(">> 5. Traversal \\n $ ");
        int choice = scan.nextInt();
        switch (choice)
        {
            case 1 :
                //Enqueue an element
                System.out.print("Enter integer element to enqueue: ");
                queue.enqueue (scan.nextInt());
                break;

            case 2 :
                //Dequeue an element
                System.out.print("You have opted to delete an element from the
queue is " );
                System.out.println(queue.dequeue());
                break;

            case 3 :

```

```

        //showing peek element
        queue.peek();
        break;

    case 4 :
        //Show size of queue
        queue.size();
        break;

    case 5 :
        //Traverse the queue
        queue.traversal();
        break;

    default :
        System.out.println("Wrong Entry \n ");
        break;
    }
    /* display stack */

    System.out.print("\nDo you want to continue (Type y or n) : ");
    ans= scan.next().charAt(0);

    } while (ans == 'Y' || ans == 'y');
    System.out.print("Thank you visit again!!!! :) " );

        scan.close();}
}

```

Question 4: Write the function transforming a decimal number into a binary number by using stack

Answer:-

```

import java.util.*;
class Node
{
    Node next;
    int data;
}

```

```

public Node()
{
    this.next = null;
    this.data = 0;
}

void set_Data(int data)
{
    this .data = data;
}
void set_Link(Node next)
{
    this.next = next;
}
int get_Data()
{
    return data;
}
Node get_Link()
{
    return next;
}
}

class Stack_Implement
{
    Node top = null;

    void push(int data)
    {
        Node node = new Node();
        if(top==null)
        {
            top = node;
            node.set_Data(data);
        }
        else
        {
            node.set_Link(top);
            node.set_Data(data);
            top = node;
        }
    }
}

```

```

int pop()
{
    int element;
    if(top==null)
    {
        System.out.println("Underflow");
        element = -1;
    }
    else
    {
        element = top.get_Data();
        top = top.next;
    }
    return element;
}

}

public class Question_4 {
    String binary(int decimal)
    {
        String binary = "";
        int size=1;
        Stack_Implement stack = new Stack_Implement();

        if(decimal>0)
        {
            while(decimal!=1)
            {
                int temp;
                temp = decimal%2;
                stack.push(temp);
                decimal = decimal/2;
                size++;
            }
            stack.push(1);

            for(int i=0;i<size;i++)
            {
                binary = binary + Integer.toString(stack.pop());
            }
        }
        else
        {
            binary = "0";
        }
    }
}

```

```

        return binary;
    }

    public static void main(String... args)
    {
        Scanner scan = new Scanner (System.in);
        Question_4 conversion = new Question_4();
        System.out.println("Enter the decimal number");
        System.out.print(conversion.binary(scan.nextInt()));

        scan.close();}
    }

```

Question 5: Write the function that removes all even numbers from the given stack. The mutual order of odd numbers must stay unchanged. The function returns the number of removed numbers.

Answer:-

```

import java.util.*;

public class Question_5 {

    private int stack[] ;
    private int size ;
    private int top ==-1;

    void set_size(int size)
    {
        this.size = size;
        stack = new int[size];
    }

    void push(int data)

    {
        if(top==size-1)

```

```

        {
            top =0;
            stack [top] = data;
        }
        else if(top==size-1)
        {
            System.out.println("Overflow");
        }
        else
        {
            top++;
            stack[top] = data;
        }
    }

    int pop()
    {
        int element=0;
        if(top==-1)
        {
            System.out.println("Underflow");
        }
        else
        {
            element = stack[top];
            top--;
        }
        return element;
    }
}

```

```

void remove_even()
{
    int i=0;
    int count = stack.length-1;
    int remove_count=0;
    int [] temp = new int[stack.length];
    int [] remove = new int [stack.length];
    while(count!=-1)
    {
        if(Math.abs(stack[count])%2==0)
        {
            remove[remove_count] = pop();

```

```

        remove_count++;
        count--;
    }
    else
    {
        temp[i] = pop();
        i++;
        count--;
    }
}

for(int j=i-1;j>-1;j--)
{
    push(temp[j]);
}
for(int j=0;j<remove_count;j++)
{
    System.out.print(remove[j]+" ");
}
System.out.println();

}

void traversal ()
{
    for(int i=top;i>-1;i--)
    {
        System.out.print(stack[i]+ " ");
    }
}

public static void main(String[] args)
{

    Scanner scan = new Scanner(System.in);
    System.out.print("Enter the size of stack:");
    Question_5 a = new Question_5();
    int size = scan.nextInt();
    a.set_size(size);
    System.out.println("Enter the elements in stack:");

```



```

        for(int i=1;i<size+1;i++)
        {
            a.push(scan.nextInt());
        }

        a.remove_even();
        a.traversal();

    scan.close();}
}

```

Question 6: Write the function that returns duplicate stack of the given stack. Duplicate stack contains the same elements as the original stack, and in the same order. The original stack must stay unchanged.

Answer:-

```

import java.util.*;

public class Question_6 {

    int top=-1;
    int [] stack;
    int size ;

    public Question_6(int size)
    {
        stack = new int[size];
        this.size = size;
    }

    void push(int data)
    {
        if(top==size-1)
        {
            top=0;

```

```

        stack[top] = data;
    }
    else if(top ==size-1)
    {
        System.out.print("Overflow");
    }
    else
    {
        top++;
        stack[top] = data;
    }
}

int pop()
{
    int element =0;
    if(top===-1)
    {
        System.out.println("Underflow");
    }
    else
    {
        element = stack[top];
        top--;
    }
    return element;
}

void traversal()
{
    for(int i=top;i>-1;i--)
    {
        System.out.print(stack[i]+" ");
    }
    System.out.println();
}

void duplicate_Stack(int []stack )
{
    Question_6 stack_2 = new Question_6(stack.length);
    Question_6 stack_3 = new Question_6(stack.length);
    for(int i=top;i>-1;i--)
    {
        stack_2.push(stack[i]);
    }
}

```

```

    }
    for(int i=0;i<stack_2.size;i++)
    {
        stack_3.push(stack_2.pop());
    }

    stack_3.traversal();
}

```

```

public static void main(String... args)

{
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter the size of stack");
    int size = scan.nextInt();
    System.out.println("Enter the elements: ");
    Question_6 stack = new Question_6(size);
    for(int i=0;i<size;i++)
    {
        stack.push(scan.nextInt());
    }
    System.out.println(stack.stack);
    stack.duplicate_Stack(stack.stack);

    scan.close();}

```

Question:7

For each of the use cases below pick a data structure (from the ones you have seen in the lecture) that is best suited for that use case, and explain your choice. If you think there is more than one suitable data structure, briefly discuss the trade-offs.

- * You want to store the stations of public transportation line. New stations can be added to both ends of the line, but not between existing station. You should be able to traverse the line in both directions.

Answer

- * We can use and implement LinkedList as we can add to both the ends, but when we take singly LinkedList we will not be able to traverse from both side, we can only traverse from one side, so we can use Doubly LinkedList, which will help us to traverse from both side, as we can store both previous and next address of station. and as we know there can be many stations; and can be added at any time, which is not possible in the case of array until and unless we use dynamic array.

* You want to store a phone book, which supports looking up a phone number by name, as well as adding and removing entries.

- * In storing of a phonebook, we can use array (2D) array to store phone number and name, in this we can easily search element using Binary Search or Interpolation Search which uses formula.

$$\text{start} + \frac{\text{end} - \text{start}}{\text{arr}(\text{end}) - \text{arr}(\text{start})} * (\text{key} - \text{arr}(\text{start}))$$

Where key here is the name we are searching.

but the only constraint in use array is the deletion of a particular contact will take time as it will have to shift all the contact by 1 place.

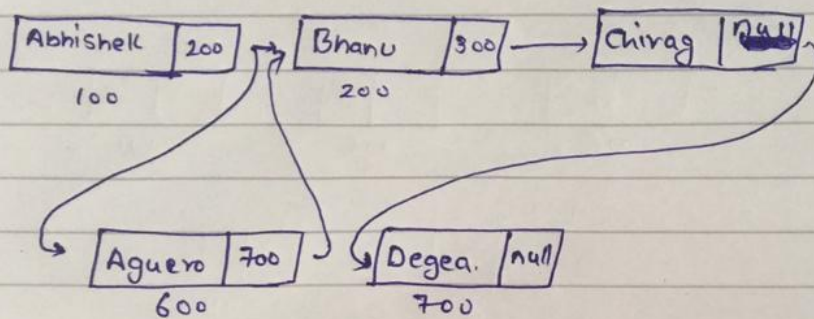
* We can also use ~~Circle~~ Doubly Linked List, because in Linked List we can ^{and delete} insert at any position easily either at start, at the end, between any contacts, but the searching the particular phone name will be difficult.

* You are looking for a way out of a maze, and you are not allowed to use recursion. You have to store the path you are currently exploring and be able to go back one step whenever you find yourself in a dead-end and exploring a new possibility from there.

* We will use stack data structure in this test case because we will be able to store our way or steps and if we find dead-end we can pop out that step from stack and go back to the previous path.

* You want to store a sorted list of string and support the operation of merging two sorted list into one, in place (without creating a copy of the lists).

* We can use linked list to merge two sorted list as if we use array we will have to move every element if we find any string smaller in the other ~~list~~ list, where as in case of linked list if we find any element smaller than the one in other linked list we simply have to link the smaller string to the ~~string~~ Node having string smaller than the string we want to add.



These both are different lists but now we are easily sorting ~~the~~ and merging the two lists.

* You are writing software for a call center. When a client calls, his calls should be stored until there is a free operator to pick up. Calls should be processed in the same order they are received.

In the test case we will use queue data structure because until the operator is free he will not be able to pick up the call that means the one who calls first will be served first and queue uses the same principle of First in First out. So we will use queue.

Question 8: A palindrome is a phrase that reads the same forward and backward (examples: 'racecar', 'radar', 'noon', or 'rats live on no evil star'). By extension we call every string a palindrome that reads the same from left to right and from right to left. Develop a recursive algorithm that takes as input a string and decides whether the string is a palindrome. Write down your algorithm in pseudocode.

Answer:-

```
palindrome (char[] S, left, right)
    Start of function
    if (left >= right)
        start of if condition
        return null or nothing
    end of if condition.
    temp = S[left]
    S[left++] = S[right]
    S[right--] = temp

    palindrome (S, left, right)

end of function.
```

At This we return nothing but after calling this function we can see that the array of characters is same as the string.

Question 9: Write a Java program to remove the duplicate elements of a given array and return the new length of the array. Sample array: [20, 20, 30, 40, 50, 50, 50]. After removing the duplicate elements, the program should return 4 as the new length of the array.

Answer:-

Assuming that the given array will always be sorted.

```
import java.util.*;
```

```
public class Question_9 {
    public int removeDuplicates(int[] arr) {
        int i=0;
        int j = 1;
        int count =0;

        if(arr.length==0) return 0;

        while(j<=arr.length-1-count)
        {
            if(arr[i]==arr[j])
            {
                for(int k=j;k<arr.length-1-count;k++)
                {
                    arr[k] = arr[k+1];
                }
                count++;
            }
            else
            {
                i++;
                j++;
            }
        }
        return arr.length-count;
    }
}
```



```

public static void main(String ... args)
{
    Scanner scan = new Scanner(System.in);

    System.out.println("Enter the size of array!!");
    int[] arr = new int[scan.nextInt()];
    System.out.println("Enter the elements: ");
    for(int i=0;i<arr.length;i++)
    {
        arr[i] = scan.nextInt();
    }
    Question_9 a = new Question_9();
    System.out.print(a.removeDuplicates(arr));
    scan.close();}
}

```

Question 10: Write a Java program for Matrix multiplication of two matrices having different sizes?

Answer:-

```

import java.util.Scanner;

public class Question_10 {

    static void matrix_multiplication(int row1, int column1, int[][] mat1,int row2, int column2, int[][] mat2)
    {
        int mat_1_row = row1, mat_1_column = column1;
        int mat_2_row = row2, mat_2_column = column2;

        int[][] firstMatrix = mat1;
        int[][] secondMatrix = mat2;

        // Mutlplying Two matrices
        int[][] product = new int[mat_1_row][mat_2_column];
        for(int i = 0; i < mat_1_row; i++) {
            for (int j = 0; j < mat_2_column; j++) {
                for (int k = 0; k < mat_1_column; k++) {
                    product[i][j] += firstMatrix[i][k] * secondMatrix[k][j];
                }
            }
        }
    }
}

```

```

        // Displaying the result
        System.out.println("Sum of two matrices is: ");
        for(int[] row : product) {
            for (int column : row) {
                System.out.print(column + "   ");
            }
            System.out.println();
        }
    }

    public static void main(String ... args)
    {

        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the size of row of matrix1: ");
        int row_1 = scan.nextInt();

        System.out.print("Enter the size of row of matrix1: ");
        int col_1 = scan .nextInt();

        System.out.println("Enter the elements into matrix");
        int [][] mat_1 = new int[row_1][col_1];

        for(int i=0;i<row_1;i++)
        {
            for(int j=0;j<col_1;j++)
            {
                mat_1[i][j] = scan.nextInt();
            }
        }

        while(true) {
            System.out.print("Enter the size of row of matrix2: ");
            int row_2 = scan.nextInt();
            System.out.print("Enter the size of row of matrix1: ");
            int col_2 = scan .nextInt();
            int [][] mat_2 = new int[row_2][col_2];

            if(col_1==row_2)
            {
                System.out.println("Enter the elements into matrix");

                for(int i=0;i<row_2;i++)
                {
                    for(int j=0;j<col_2;j++)
                    {
                        mat_2[i][j] = scan.nextInt();
                    }
                }
            }
        }
    }
}

```

```
matrix_multiplication(row_1, col_1, mat_1, row_2, col_2, mat_2);

                break;
    }

    else
    {
        System.out.println("Size of matrix for multipilcation is not
correct!!!(column of matrix 1 should be equal to rowof matrix 2");
        System.out.println("Enter the size again");
        }
        scan.close();
    }

}
```