Hacettepe University

BBM104 Introduction to Programming Laboratory II

Assignment IV Report

Name : Mustafa Surname : KOLLU Number : 21627485

Problem Definiton

In this assignment what we are expected write 6 function for stack and queue like calculateDistance . We need to solve this problem with write own stack and queue methods like push,pop,enqueue (implementation). We are expected to our program work with just stack and queue methods.

Solution

- I write expected function like pop,push,isFull & isEmpty for stack.

```
public static boolean isEmpty(){                    //isEmpty() function
    if(top==-1) {
        return true;
    }
    return false;
}

public static boolean isFull() {                    //isFull() function
    if(top==99) {
        return true;
    }
    return false;
}

public static void push(int x) {                    //push function
    if(isFull()==true) {
        System.out.println("Stack contains a maximum of 100 elements");
    }
    else {
        ++top;
        stack[top]=x;
    }
}
```

```
public static int pop() {
    if(isEmpty()==true) {
        System.out.println("Stack is Empty1");
    }
    else {
        int x=stack[top];
        stack[top]=x;
        top--;
        return x;
    }
    return top;
}
```

- I write expected function like enqueue,push,isFull & isEmpty for stack.

```
public static boolean isEmpty(){                    //isEmpty() function
    if(rear==-1) {
        return true;
    }
    return false;
}

public static boolean isFull() {                    //isFull() function
    if(rear==99) {
        return true;
    }
    return false;
}

public static void enqueue(int x) {                 //enqueue function
    if(isFull()==true) {
        System.out.println("Queue contains a maximum of 100 elements");
    }
    else {
        ++rear;
        queue[rear]=x;
    }
}
```

```
public static int dequeue() {
    if(isEmpty()==true) {
        System.out.println("Queue is Empty");
    }
    else {
        int x=queue[front];
        queue[front]=x;
        front++;
        return x;
    }
    return front;
}
```

- Then I used this function for our real problem(command function) like removeGreater.

```java
public static void removeGreater(int k) {
    int temp2=top2;
    for(int j=0;j<=temp2;j++) {
        pushStack3(pop2());
    }
    int temp3=top3;
    for(int j=0;j<=temp3;j++) {
        if(stack3[temp3-j]<=k) {
            push(pop3());
        }
        else {
            pop3();
        }
    }
}
```

```java
public static void removeGreater(int k) {
    int temp2=rear2;
    int temp4=front2;
    for(;temp4<=temp2;temp4++) {
        if(queue2[front2]<=k) {
            enqueue(dequeue2());
        }
        else {
            dequeue2();
        }
    }
    rear2=-1;
    front2=0;
}
```

1. Command Class
   This class just reads and hold elements of command.txt

2. Main Class
   This class just run work function of Program class

3. Program Class
   This class run function of Stack class and Queue class by commands.

4. Queue Class
   This class have enqueue,dequeue,isFull,isEmpty implemantation and Command function like removeGreater.

5. Stack Class
   This class have push,pop,isFull,isEmpty implemantation and Command function like removeGreater.

Analysis Of Commands

- Complexity describes the amount of time it takes to run an algorithm.

- The highest complexity among functions is O (n ^ 2) in queue class and stack class.

```java
for(;temp1<=temp;temp1++) {
    int counter=0;
    int temp8=dequeue();
    enqueue2(temp8);
    int temp3=rear3;
    int temp6=front3;
    for(;temp6<=temp3;temp6++) {
        int temp99=dequeue3();
        if(temp8==temp99) {
            counter++;
            enqueue4(temp99);
        }
        else {
            enqueue4(temp99);
        }
    }
    rear3=-1;
    front3=0;
    if(counter>=1) {
        total++;
    }
    int temp9=rear4;
    int temp10=front4;
    for(;temp10<=temp9;temp10++) {
        int temp88=dequeue4();
        if(temp8==temp88) {
            continue;
        }
        else {
            enqueue3(temp88);
        }
    }
    rear4=-1;
    front4=0;
```

```java
for(int i=0;i<=temp1;i++) {
    int temp5=pop();
    pushStack2(temp5);
    int temp3=top3;
    for(int j=0;j<=temp3;j++) {
        int temp6=pop3();
        total+=Math.abs(temp5-temp6);
    }
    int temp7=top;
    for(int j=0;j<=temp7;j++) {
        int temp8=pop();
        pushStack2(temp8);
        pushStack3(temp8);
    }
    int temp9=top3;
    for(int j=0;j<=temp9;j++) {
        push(pop2());
    }
```