

Lab 3

| Function | Big O |
|--|--|
| <pre>int*queue;//the array that holds the queue int queue_size;//size of the queue[] array int opCode; int front=0, rear=0;//queue index variables</pre> | <p>O(1) [declaration] O(1)</p> |
| <pre>int queueSize() { int size; size=rear-front; return size; }</pre> | <p>O(1)[declaration] O(1)[arithmetic, assignment] O(1)[return] O(1)</p> |
| <pre>int isEmptyQueue() { if (front==rear) {return 1;} else return 0; }</pre> | <p>O(1)[comparison] O(1)[return] O(1)</p> |
| <pre>int isQueueFull() { if(rear==queue_size) { return 1; } else return 0; }</pre> | <p>O(1)[comparison] O(1)[return] O(1)</p> |
| <pre>int enqueue(int x) { if(rear==queue_size) { return -1; } else { queue[rear]=x; rear=rear+1; return 1;} }</pre> | <p>O(1)[comparison] O(1)[return] O(1)[index, assignment] O(1)[arithmetic, assignment] O(1)</p> |
| <pre>int dequeue() { int x; if(front==rear) { return -1; } else { x=queue[front]; front=front++; return x;} }</pre> | <p>O(1)[declaration] O(1)[comparison] O(1)[return] O(1)[index, assignment] O(1)[increment]</p> |

| | |
|---|--------|
| } | $O(1)$ |
|---|--------|

Average running time



Ques 2:

```

Telnet 172.16.22.5
linuxbpd1 login: 2019A7PS0020U
Password:
Last login: Wed Apr  7 12:43:34 from ::ffff:10.30.17.128
[2019A7PS0020U@linuxbpd1 ~]$ cd DSALab-2021/Week_03
[2019A7PS0020U@linuxbpd1 Week_03]$ ls
SearchTestcase.txt  a.out  lab3_q2.c  queue.c  queueTime.txt  testCaseSize.txt
[2019A7PS0020U@linuxbpd1 Week_03]$ cc lab3_q2.c
[2019A7PS0020U@linuxbpd1 Week_03]$ a.out

1: Enqueue element into queue
2: Dequeue element from queue
3: Exit
Enter choice: 2
Error, stack is empty
Enter choice: 1
Enter integer to enqueue: 2
2 enqueued successfully
Enter choice: 1
Enter integer to enqueue: 3
3 enqueued successfully
Enter choice: 2
2 dequeued successfully
Enter choice: 2
3 dequeued successfully
Enter choice: 2
Error, stack is empty
Enter choice: 3
Exiting...
Exited
[2019A7PS0020U@linuxbpd1 Week_03]$

```

| Function | Big O |
|--|-----------------------------|
| <pre> int isStack2Full() { if(top2==stack_size-1) return 1; else return 0; } </pre> | $O(1)$ [comparison, return] |
| <pre> int isStack2Empty()//returns 1 if stack2[] is empty and 0 otherwise { if(top2==-1) return 1; else return 0; } </pre> | $O(1)$ [comparison, return] |
| <pre> int pop2() { int data; if(!isStack2Empty()) { </pre> | $O(1)$ [declaration] |

| | |
|--|---|
| <pre> data = stack2[top2]; top2 = top2 - 1; return data; } else return -1; } int push2(int x) { if(!isStack2Full()) { top2 = top2 + 1; stack2[top2] = x; return 1; } else return -1; } </pre> | <p> $O(1)$[index, assignment] $O(1)$[arithmetic, assignment] $O(1)$[return] </p> <p> $O(1)$[arithmetic, assignment] $O(1)$[index, assignment] $O(1)$[return] </p> <p>$O(1)$</p> |
| <pre> int deQueue() { if(isQueueEmpty()) {printf("Error, stack is empty"); return -1;} else { while (isStack2Empty() == 0) { int x_1=pop2(); push1(x_1); } int y=pop1(); while (isStack1Empty() == 0) { int x_2=pop1(); push2(x_2); } printf("%d dequeued successfully",y); return y; } } </pre> | <p> $O(n)$[till end of list] </p> <p> n n </p> <p> $O(1)$[pop func, assignment] $O(n)$[till end of list] </p> <p> n n </p> <p>$O(n)$</p> |