

# Birla Institute of Technology and Science, Pilani, Hyderabad Campus

Department of Computer Sc. and Information Systems

CS F211 (Data Structures and Algorithms)

Summer Term 2022

Lab Sheet 4



**BITS Pilani**  
Hyderabad Campus

Date: 13<sup>th</sup> June 2022

## Deque ADT

A deque is somewhat of a hybrid between a stack and a regular queue that supports operations to both add and remove elements from the front and back of the queue. Since we likely have to perform a lot of operations on both ends of the deque, it will probably be beneficial to use a doubly-linked list over a singly-linked one.

Structure for creating deque using doubly linked list is as follows

```
struct dqnode
{
    int data;
    struct dqnode *next;
    struct dqnode *prev
};

struct deque
{
    struct dqnode *front, *rear;
};
```

**Task 1: Implement the following operations of deque.**

**Operations:**

• **int size ()**

Requirements: None.

Results: returns the number of elements in the deque.

• **bool Isempy()**

Requirements: None.

Results: returns true if the deque is empty, else returns false.

• **int front()**

Requirements: List is not empty.

Results: returns the value of the element at the front of the deque.

• **int back()**

Requirements: List is not empty.

Results: returns the element at the back of the deque.

•**void addFront(int e)**

Requirements: None

Results: adds a new element to the front of the deque.

•**void addBack(int e)**

Requirements: None

Results: adds a new element to the back of the deque.

•**void removeFront()**

Requirements: List is not empty

Results: removes the element at the front of the deque.

•**void removeBack()**

Requirements: List is not empty

Results: removes the element at the back of the deque.

•**void listReverse()**

Requirements: List is not empty

Results: reverses the order of elements in the deque.

**Task 2:**

The stock span problem is a financial problem where we have a series of  $n$  daily price quotes for a stock and we need to calculate span of stock's price for all  $n$  days. The span  $S_i$  of the stock's price on a given day  $i$  is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than its price on the given day (this is including the current day also). For example, if an array of 7 days prices is given as 100, 80, 60, 70, 60, 75, 85, then the span values for corresponding 7 days are 1, 1, 1, 2, 1, 4, 6. Given an array of daily stock prices for  $n$  days, you have to find the stock span for each day. You must accomplish this in  $O(n)$  running time, where  $n$  is the number of days.

**void findStockSpan (int n, int a[])**

Requirements: None

Results: should output a list of all the stock spans for each day. There must be a total of  $n$  space separated numbers with a newline printed after them.

**Hint:** For this task, try using a stack to come up with a fast algorithm.

**Prefix expression:**

An expression is called the prefix expression if the operator appears in the expression before the operands. Example :  $*+AB-CD$  (Infix :  $(A+B) * (C-D)$  )

**Task 3:**

Write a program to evaluate a prefix expression using a single stack. You can use functions defined in the the deque implementation for implementing stack operators or you can use code from the previous lab.

**Sample input1:** + 3 \* + 5 9 2

**Output:** 31

**Sample input 2:** + \* 5 4 - 100 / 20 2

**Output:** 110