FACE Prep

Minimum Stack

Minimum Stack

Problem: Design and implement a stack that supports push(),pop(), top() and retrieving the minimum element in constant time.

Implement a Stack class, which supports the following methods in O(1) time complexity.

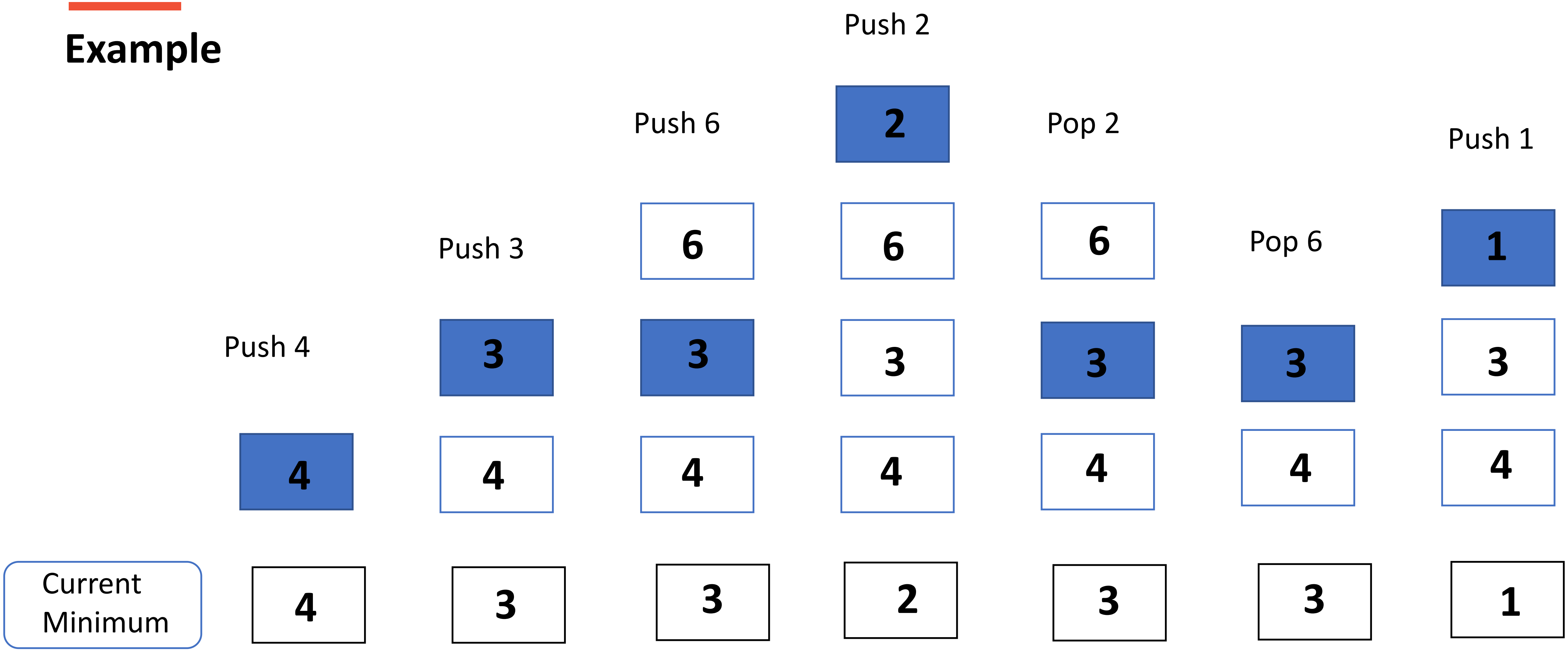
void push() : Insert element onto the stack.

void pop() : Remove the top element from the stack.

int top() : Retrieve the top element in the stack.

int getmin() : Retrieve the minimum element in the stack.

Example



	Push 4	Push 3	Push 6	Push 2	Pop 2	Pop 6	Push 1
				2			
			6	6	6		1
		3	3	3	3	3	3
	4	4	4	4	4	4	4
Current Minimum	4	3	3	2	3	3	1

```java
import java.util.*;
class Mystack {
        Stack<Integer> s;
        Stack<Integer> a;
        Mystack() {
                s = new Stack<Integer>();
                a = new Stack<Integer>();
        }
        void getMin() {
                if(a.isEmpty())
                        System.out.println(“Stack is Empty”);
                else
                        System.out.println(“Minimum element : “ + a.peek());
        }
        void peek(){
                if(s.isEmpty()) {
                        System.out.println(“Stack is Empty”);
                        return ;
                }
```

```
       integer t=s.peek();
       System.out.print(“Top most element:” + t);
    }
    void pop() {
       int t = s.pop();
       if(s.isEmpty()) {
             System.out.println(“Stack is Empty”);
             return ;
       }
       else
             System.out.println(“Removed element : “ + t);
       if(t == a.peek())
             a.pop();
    }
```

```
void push(int x) {
   if ( s.isEmpty()) {
         s.push(x);
         a.push(x)
         System.out.println(“ Number Inserted: “+ x);
         return ;
   }
   else {
         s.push(x);
         System.out.prinln(“ Number Inserted: “ +x);}
   if ( x<= a.peek() )
         a.push(x);
   }
};
```

```java
public class Main {
    public static void main(String args[]) {
        Mystack s=new Mystack();
        Scanner sc = new Scanner(System.in);
        int n=sc.nextInt();
        for( int i=0;i<n;i++) {
            int m=sc.nextInt();
            s.push(m);
        }
    s.getMin();
    s.pop();
    s.getMin();
    s.pop();
    s.peek();
    }
}
```

THANK YOU