

MAHARISHI INTERNATIONAL UNIVERSITY

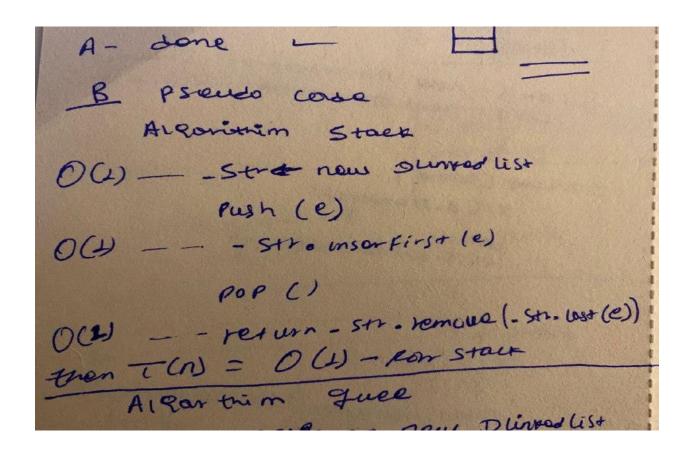
Assignment two



Implement in JavaScript the function find Middle (L) than we did in class. The DLinkedList class is provided in the DLinkedList.js file.

A. Describe, in pseudo-code, how to implement the stack ADT using a DLinkedList. What is the running time of the push () and pop () methods in this case? Implement a new Stack class in JavaScript based on (using) the DLinkedList class like done in A above.

Pseudo code figure



Source code

```
■ DLinkedList (1).js ×
                                                            ▶
 228
 229 class stack {
 230
         constructor() {
 231
             this._str = new DLinkedList()
 232
 233
          push(e) {
 234
             this._str.insertFirst(e)
 235
          }
 236
          pop() {
 237
             return this. str.remove(this. str.first())
 238
          }
 239
          out() {
 240
             return this._str.print();
 241
 242 }
 243
 244 let stkobj = new stack()
 245 stkobj.push(23)
 246 stkobj.push(33)
 247 stkobj.push(43)
 248 stkobj.out();
 249 stkobj. pop()
 250 console.log("stack after pop")
 251 stkobj.out();
```

B. Describe, in pseudo-code, how to implement the queue ADT using a DLinkedList. What is the running time of the enqueue() and dequeue() methods in this case? Implement a new Queue class in JavaScript based on the DLinkedList class.

Pseudo code for queue

Source code

```
let queobj=new quee()

queobj.enquee(12);

queobj.enquee(32);

queobj.enquee(122);

queobj.printout();

queobj.deque();

queobj.printout();

queobj.printout();
```

C. C-2.2 Describe, in pseudo-code, how to implement the queue ADT using two stacks. What is the running time of the enqueue() and dequeue() methods in this case?

Pseudo code with running time

E.2.2 Algorithim anostack Free Input: any liabil data autput: List mbax now Stack authors new Stack ZP (enbox - is empth 7) then - O(4) enque (e) mbox · push (e) - O(4) tonau ("on error") - 0 4) eise Tin) -> 0(4) deque () O(4) - IF (out box = iremety)) O(m) - mule (! inbox. emoteu) O(1) - Outbox. Push (inbox. POP (1) - return out box. POP (); TON = O(n) entue con core 0(1) de fue con Tore O(n)