MAHARISHI INTERNATIONAL UNIVERSITY

Assignment two

OCTOBER 30, 2020

KASAHUN TEHONE

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

```javascript
  function findMiddle(elem) {

    // let p = this._header.next;
    // let q = this._trailer.prev;

     let p = elem.first();
    let q = elem.last();
    if (elem.isEmpty()) {
        throw new Error ("nod middle")
        // p = elem.first();
        // q = elem.last();
    }
    while (p != q && elem.after(p) != q)
    p = elem.after(p);
     q = elem.before(q);
    return q._elem;
}

console.log("THE MIDDLE ELEMENT",findMiddle(tst2))
```

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

A- done

B pseudo code

Algorithim Stack

O(1) — -Str new DLinked list

Push (e)

O(1) — — - Str. insertFirst (e)

pop ()

O(1) — - return - str. remove (- str. last (e))

then T(n) = O(1) - for stack

Algorithim Queue

new DLinked List

Source code

DLinkedList (1).js

```
class stack {
    constructor() {
        this._str = new DLinkedList()
    }
    push(e) {
        this._str.insertFirst(e)
    }
    pop() {
        return this._str.remove(this._str.first())
    }
    out() {
        return this._str.print();
    }
}

let stkobj = new stack()
stkobj.push(23)
stkobj.push(33)
stkobj.push(43)
stkobj.out();
stkobj.pop()
console.log("stack after pop")
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

Algorithm Queue

O(1) ---- queue ← new DLinkedList

Enqueue (e)

O(1) ---- queue. insertLast (e)

dequeue ()

O(1) -- return queue. remove (queue. First())

then T(n) = O(1) – for Queue Implimentation.

Source code

```
class quee
{
    constructor()
    {
        this._queerow=new DLinkedList();
    }
    enquee(e)
    {
        this._queerow.insertLast(e)
    }
    /**
     *
     * @param {object or any data}  e
     * @return {array or llinked lisr}
     */
    deque()
    {
        return this._queerow.remove(this._queerow.first())
    }
    printout()
    {
        return this._queerow.print();
    }
```

```
let queobj=new quee()
queobj.enquee(12);
queobj.enquee(32);
queobj.enquee(122);
queobj.printout();
queobj.deque();
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

Algorithm TwoStackQueue

input: any valid data
output: List

inbox ← new Stack
outbox ← new Stack

enque (e)
IF (inbox.isempty) then — O(1)
inbox.push (e) — O(1)

else
throw ("an error") — O(1)
T(n) → O(1)

deque ()
O(1) — IF (outbox.isempty())
O(n) — while (! inbox.empty())
O(n) — outbox.push (inbox.pop())
O(1) — return outbox.pop();

T(n) = O(n)

enque con torre O(1)
deque con torre O(n)