

**WIA2005 Algorithm Design And Analysis**  
Faculty of Computer Science and Information Technology  
University of Malaya

**REPORT 2**

(Lab 7 – Lab 11)

**Prepared by**

|  |  |
| --- | --- |
| Name | Matrix No. |
| MUHAMMAD AMIRUL BIN HARUN | WES150016 |
| MUHAMMAD FARIS BIN JAAFAR | WEM150013 |
| MUHAMMAD AZRI BIN ZAINUDDIN | WER150018 |
| IRSYAD BIN SHAMSUDIN | WES150009 |

Lecturer’s Name

Dr. Asmiza binti Abdul Sani

Contents

[Lab 1 3](#_Toc478892728)

[Part A 3](#_Toc478892729)

[Question 1 3](#_Toc478892730)

[Question 2 3](#_Toc478892731)

[Question 3 3](#_Toc478892732)

[Question 4 3](#_Toc478892733)

[Question 5 3](#_Toc478892734)

[Question 6 3](#_Toc478892735)

[Question 7 3](#_Toc478892736)

[Part B 4](#_Toc478892737)

[Question 1 4](#_Toc478892738)

[Question 2 4](#_Toc478892739)

# Lab 7

### Question 1

Queue is a kind of abstract data type or collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position, known as enqueue, and removal of entities from the front terminal position, known as dequeue. This makes the queue a First-In-First-Out (FIFO) data structure. There are two main operations: enqueue (adding elements) and dequeue (removing elements).

**Pseudocode:**

Enqueue (Q, x)

Q [Q.tail] = x

If Q.tail == Q.length

Q.tail = 1

Else

Q.tail = Q.tail + 1

Dequeue (Q)

X = Q [Q.head]

If Q.head == Q.length

Q.head = 1

Else

Q. head = Q.head + 1

Return x

Running time:

* Enqueue: O (1)
* Dequeue: O (1)

### Question 2

Stack is an abstract data type that serves as a collection of elements, with two principal operations: push, which adds an element to the collection, and pop, which removes the most recently added element that was not yet removed. The order in which elements come off a stack gives rise to its alternative name, Last-In, First-Out (FIFO).

A stack can be implemented by using a linked list or an array.

**Pseudocode:**

Push (S, x)

S.top = S.top + 1

S[S.top] = x

Pop (S)

If Stack-Empty(S)

Error “underflow”

Else

S.top = S.top -1

Return S [S.top + 1]

Running time:

* Push: O (1)
* Pop: O (1)

### Question 3

Queue with two stacks is a form of queue implemented with two stacks as basic operation.

**Pseudocode:**

Create two stacks: 's' and 'tmp' as in the program given below

For insert operation:

if size of s = 0 then

push value into s

else

push all popped elements from s to tmp

push value into s

push all popped elements from tmp to s

For remove operation:

if size of s = 0 then

throw 'underflow' exception

else

return pop element from s

Running time:

* Enqueue: O (1)
* Dequeue:
  + O (n), when tmp is empty.
  + O (1) If we consider that each element will be in each queue exactly once, then we realize that each element will be enqueued exactly twice and dequeued exactly twice.

### Question 4

Linked list is a linear collection of data elements, called nodes, each pointing to the next node by means of a pointer. It is a data structure consisting of a group of nodes which together represent a sequence

**Pseudocode:**

List-Search (L, k)

X = L.head

While (x != NIL and x.key != k)

X = x.next

Return x

List-Insert (L, x)

x.next = L.head

if L.head != NIL

L.head.prev = x

L.head = x

x.prev = NIL

List-Delete (L,x)

If x.prev != NIL

x.prev.next = x.next

else

L.head = x.next

If x.next != NIL

x.next.prev. = x.prev

Running time:

* Searching a linked list: O (n)
* Inserting into a linked list: O (1)
* Deleting from a linked list: O (1), but if we wish to delete and element with a given key, O (n) time is required in the worst case because we must first call LIST-SEARCH to find the element.

### Question 5

Cyclic linked list is a linked list, but In the last node of a list, it points back to the first node of the list.

**Pseudocode:**

List-Insert-Cyclic (L, x)

Create the new node

Set the new node’s next to itself

If the list is empty

return new node.

Set our new node’s next to the front.

Set tail’s next to our new node.

Return the end of the list.