Lee Moriarity

Professor Kelly

CIS 112 ONL01

01 October 2022

Lab #4

Exercise #2:

2. Indicate whether a queue would be a suitable data structure to use in each of the

following applications.

a. An ailing company wants to evaluate employee records so that it can lay off

some workers on the basis of service time (the most recently hired employees

are laid off first).

No, a Stack would be a more suitable option for a Last in, First out scenario.

b. A program is to keep track of patients as they check into a clinic, assigning them

to doctors on a ”first come, first serve” basis.

Yes, Queue is most suitable for a clinic operating on a first come, first serve basis.

c. A program to solve a maze is to backtrack to an earlier position (the last place

where a choice was made) when a dead-end position is reached.

No. a Stack would be more suitable because it will store the values by last position, so the most recent location will always be on top of the stack.

d. An operating system is to process requests for computer resources by allocating

the resources in the order in which they are requested.

Yes, a Queue would be most suitable because the order which they are requested (First In First Out).

e. A grocery chain wants to run a simulation to see how average customer wait

time would be affected by changing the number of checkout lines in its stores.

Yes. Queue is most suitable for a first come first serve checkout line.

f. Customers are to take numbers at a bakery and be served in order when their

number comes up.

Yes. Since the numbers are served in order of the Queue (First In First Out) it is most suitable.

g. Gamblers take numbers in the lottery and win if their numbers are picked.

No, since the numbers will be picked randomly a Queue would not be most suitable.

Exercise #4:

4. Show what is written by the following segments of code, given that element1,

element2, and element3 are int variables, and q is an object that its the abstract

description of a queue as given in Section 4.2, “The Queue Interface.” Assume you

can store and retrieve values of type int in q.

a. element1 = 1; element2 = 0; element3 = 4; // Iniitialize variables

q.enqueue(element2); // push element2 into q

q.enqueue(element1); // push element1 into q

q.enqueue(element1 + element3); // push element1 + element3 into q

element2 = q.dequeue(); // assign first element of q to element2 and remove it from q

q.enqueue(element3\*element3); // push element3\*element3 into q

q.enqueue(element2); // push element2 back into q

q.enqueue(3); // push 3 to q

element1 = q.dequeue(); // assign first element of Queue to element1 and remove it from queue

System.out.println(element1 + " " + element2 + " " + element3); // print element1 element2 element3

q.enqueue(element3\*element3); // push element3\*element3 into q

q.enqueue(element2); // push element2 into queue

q.enqueue(3); // push 3 to queue

while (!q.isEmpty()) // while queue is not empty

{

element1 = q.dequeue(); System.out.println(element1); // print and remove each first item from the queue

}

b. element1 = 4; element3 = 0; element2 = element1 + 1; // Initialize variables

q.enqueue(element2); // push element2 to q

q.enqueue(element2 + 1); // push element2 + 1 to q

q.enqueue(element1); // push element1 to q

element2 = q.dequeue(); // assign first element of q to element2 and remove from queue

element1 = element2 + 1; // element1 equals element2 plus 1

q.enqueue(element1); // push element2 to back of queue

while (!q.isEmpty()) // while q is not empty

{

element1 = q.dequeue(); // assign element1 to first value of q and remove it from q

System.out.println(element1); // print element1

}

System.out.println(element1 + " " + element2 + " " + element3); // print values of element 1 element 2 element 3

Chart, bar chart

Description automatically generated

Exercise #33:

Using the circular doubly linked list below, give the expression corresponding to

each of the following descriptions.

Chart, diagram, box and whisker chart

Description automatically generated

For example, the expression for the info value of node 1, referenced from reference A,

would be A.getInfo().

1. The info value of node 1, referenced from reference C

C.next().getInfo()

1. The info value of node 2, referenced from reference B

B.back().getInfo()

1. The link value of node 2, referenced from reference A

A.next.getLink()

1. The link value of node 4, referenced from reference C

C.back.getBack()

1. Node 1, referenced from reference B

B.back().back()

1. The back value of node 4, referenced from reference C

C.back().getBack()

1. The back value of node 1, referenced from reference A

A.getback()