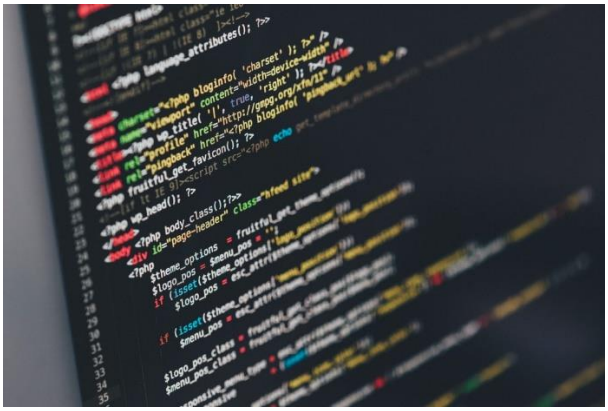
VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ASSIGNMENT

CSE2011 – DATA STRUCTURES AND ALGORITHMS



(B.Tech. CSE Specialisation in Bioinformatics)

WINTER SEMESTER 2020-2021

Name:	**ALOK MATHUR**
Reg. No:	**20BCB0086**
Slot:	**A1**
Faculty Name:	**SRIVANI A Ma'am**

VIT – A Place to Learn; A Chance to Grow

1.Applications of Stacks and Queues

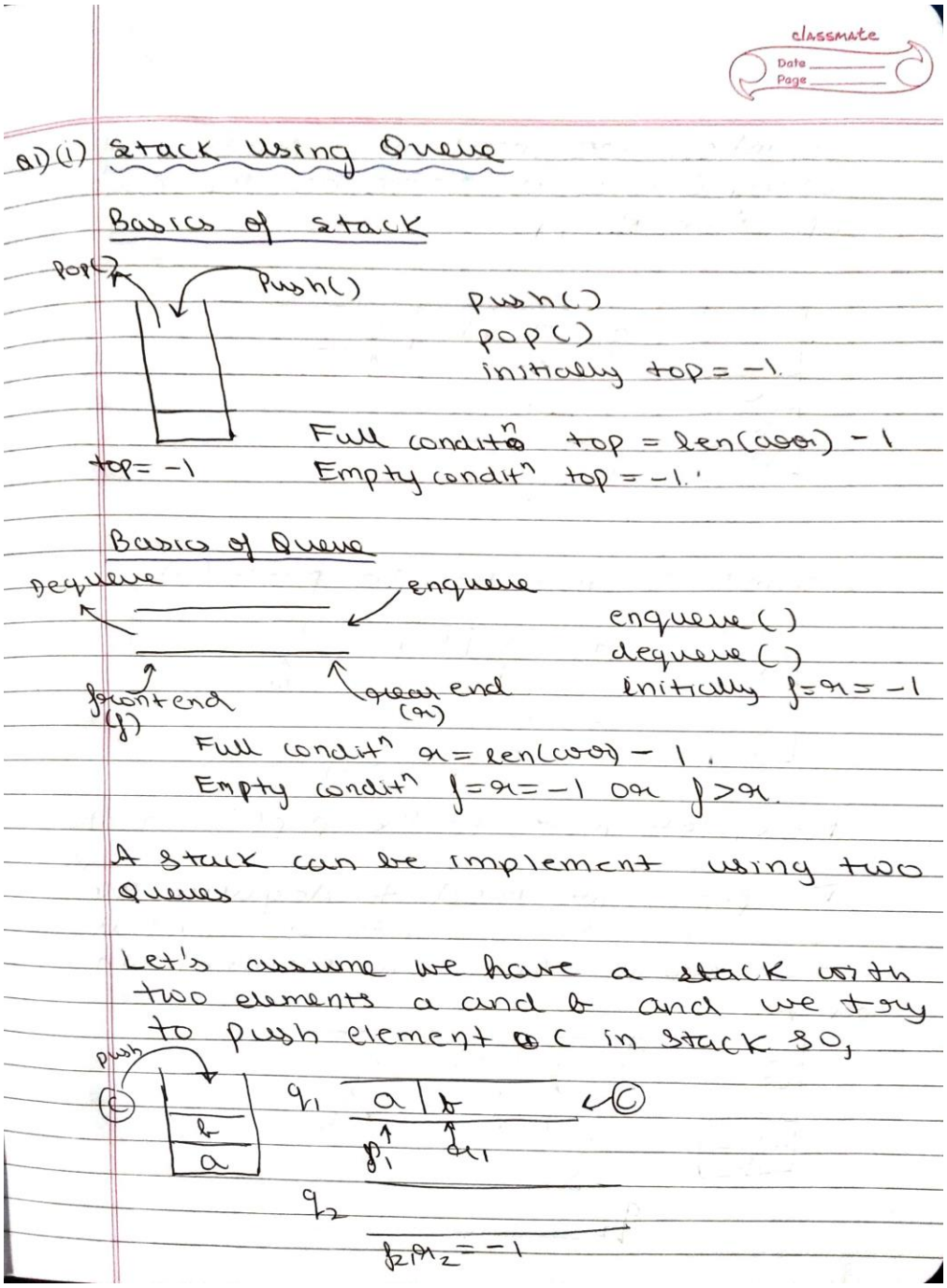
Stack using queues:

Explain in details how a stack can be implemented using two queues along with its algorithm and diagrams

Queue using stacks:

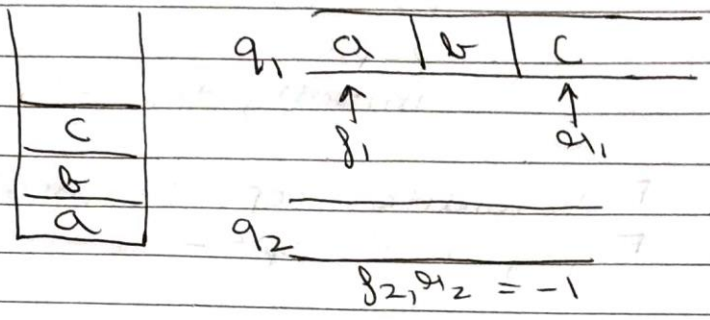
Explain in details how a queue can be implemented using two stacks along with its algorithm and diagrams

Answer

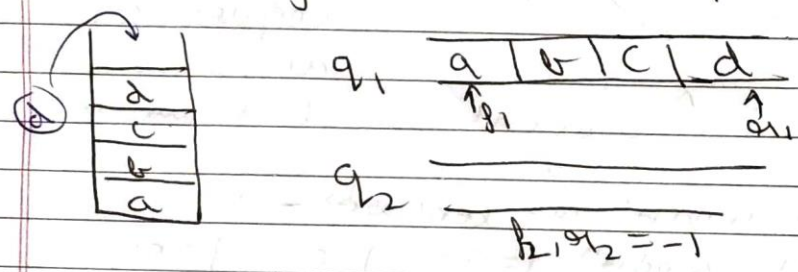


So, when we implement push() of stack using Queue its same as the enqueue operation of q_1

∴

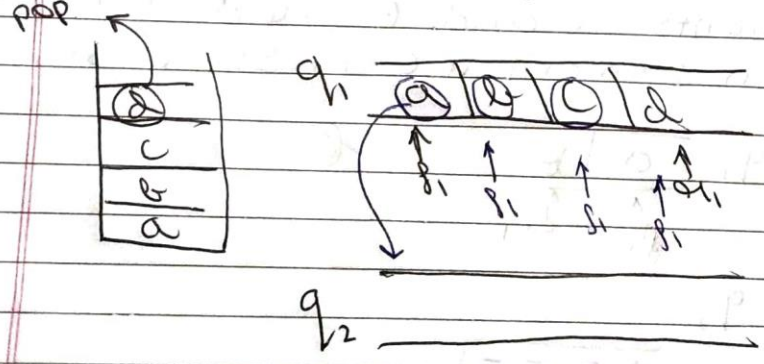


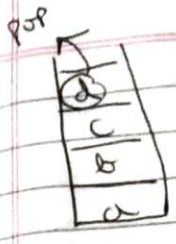
∴ Similarly now lets do push(d).

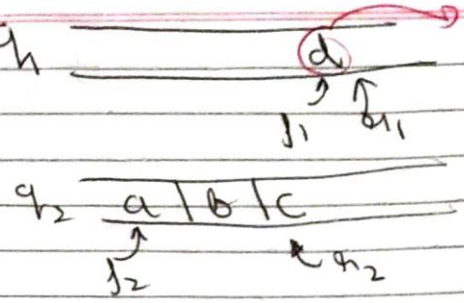


Now, lets move to the pop() operation.

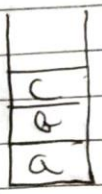
For pop() from q_1 we need to dequeue the element and store into q_2 untill there is only one element in q_1

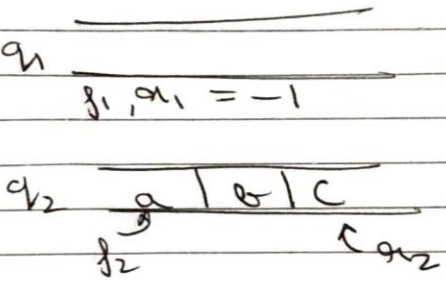






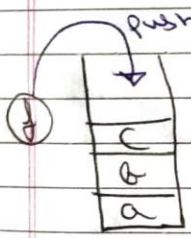
dequeue in q_1 will result into pop(d)

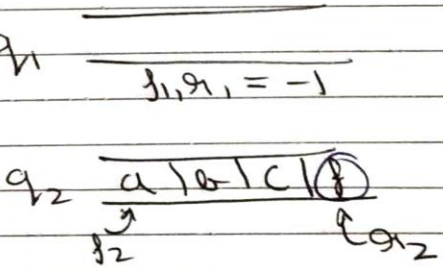




Now, similarly lets perform push(f) operation.

We will do the enqueue operation in q_2





Similarly for the pop() operation we will do it on q_2. Dequeue in q_2 and enqueuing q_1 until there is only 1 element in q_2

Algorithm

Push Algorithm

The following are the steps to perform the push operation:

Step 1: Consider two queues, i.e., Q1 and Q2, and the element to be inserted in the queue is x.

Step 2: element= Q1.enqueue(x);

Step 3: return element;

Pop Algorithm

The following are the steps to delete an element from the queue:

Step 1: Consider two queues, i.e., Q1 and Q2, and we want to remove an element from the queue.

Step 2: if !Q1.isEmpty() then

size:= Q1.size();

for i=0…size-1 do

Q2.enqueue(Q1.dequeue());

end

int item = Q1.dequeue();

for j=0…size-1 do

Q1.enqueue(Q2.dequeue());

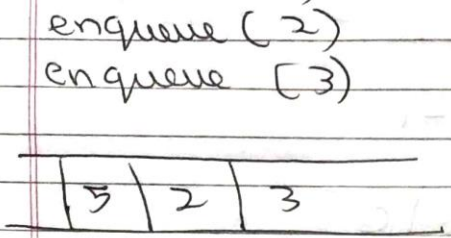
end

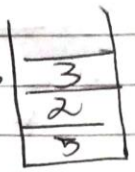
(ii) Queue Using Stack

Inorder to implement a queue using stack we will need 2 stacks.

Let assume we do the following operation

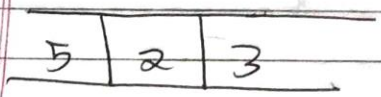
enqueue (5)
enqueue (2)
enqueue (3)

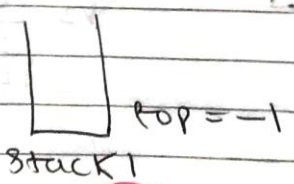


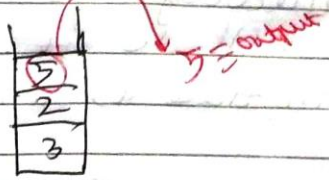


The enqueue operation is same as push in stack 1.

Inorder to dequeue() element we will ~~pop~~ push_stack2(pop_stack1()) operation.







Now, we can pop_stack2()

Algorithm

Enqueue

Step 1: Take two stacks S1 and S2.

Step 2: Push everything to S1 taking into consideration that S1 has unlimited size.

Dequeue

Step 1: If both S1 and S2 is empty return -1.

Step 2: Push everything to S2 from S1.

Step 3: Delete(pop) the top element from S2.

2. Operations on Binary Tree

i) For a binary tree what will be the result of the following. Give examples for each with neat diagrams

Left View of the tree

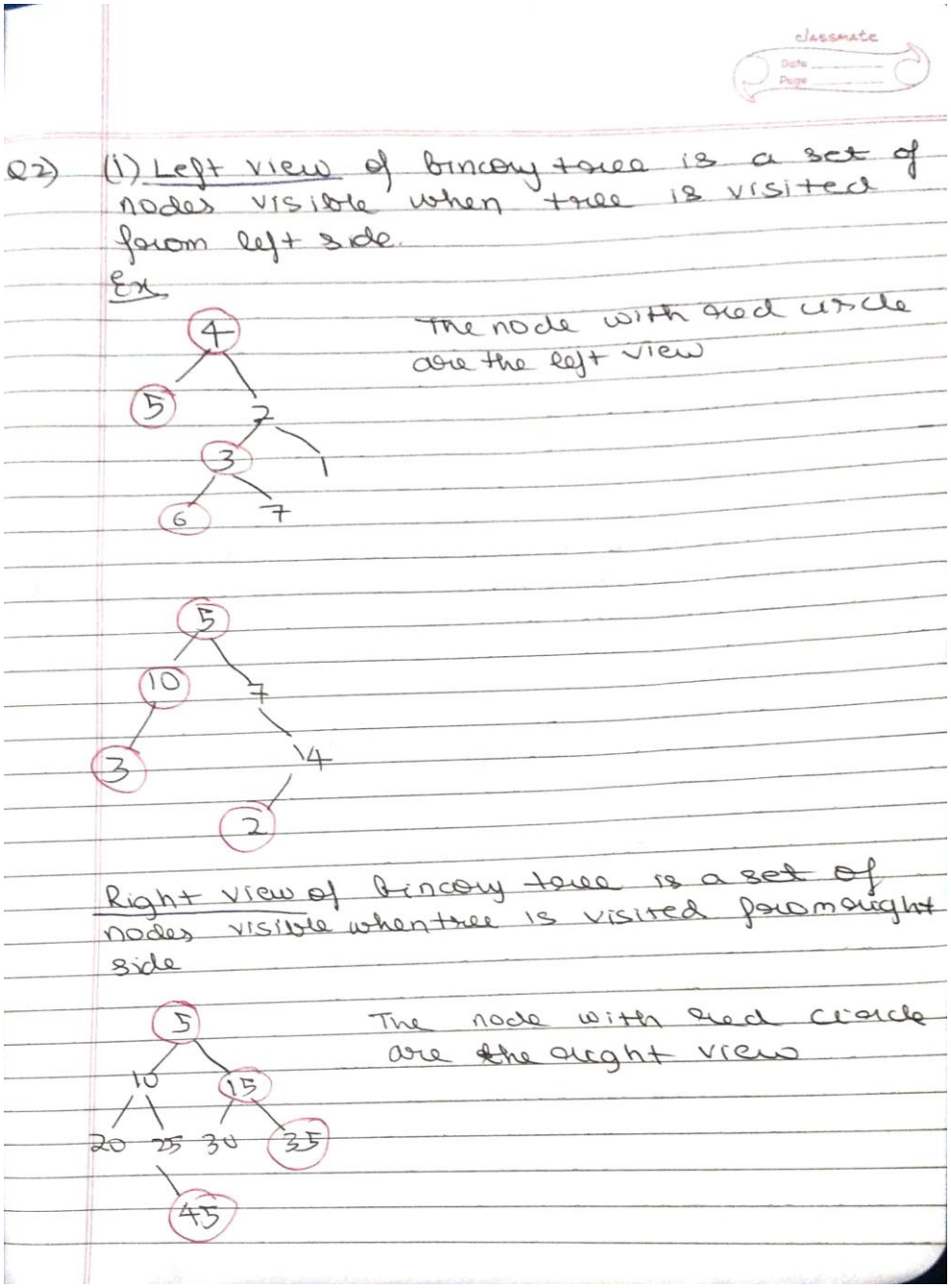
Right View

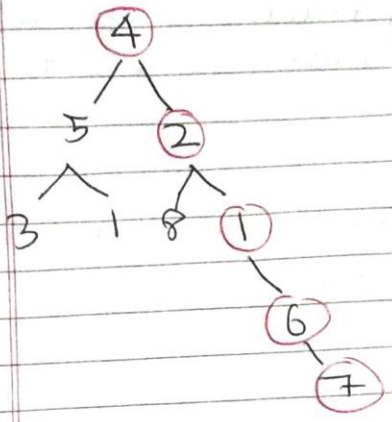
Top View

Bottom View

Mirror Image

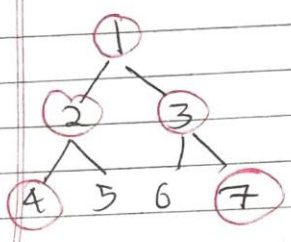
Answer



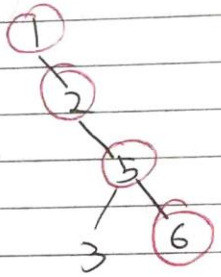


Top View of binary tree is the set of nodes visible when the tree is viewed from the top.

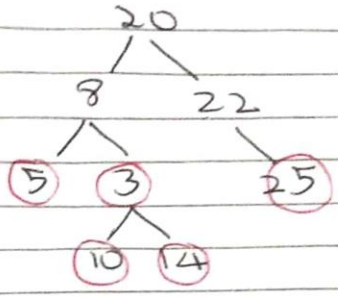
Example



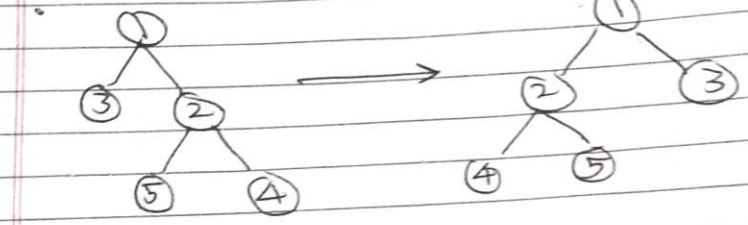
The node with red circle is top view

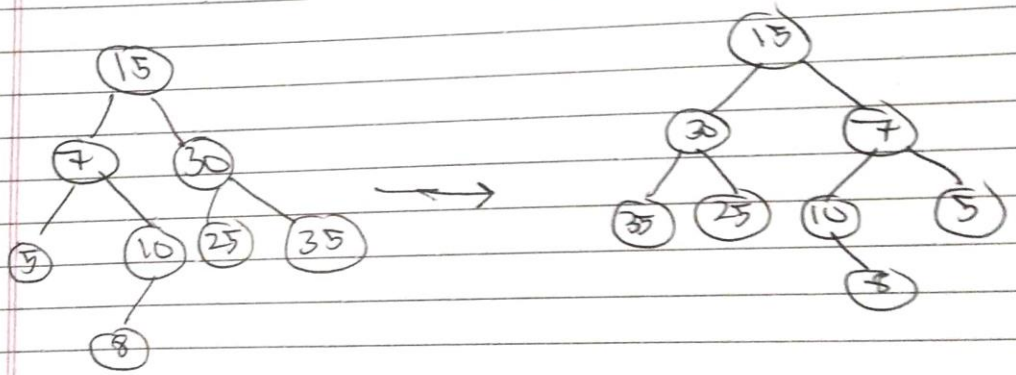


Bottom view of a binary tree refers to the bottommost nodes present at their horizontal distance.



A mirror image of a binary tree with left and right children of all non-leaf nodes of the given binary tree are interchanged.





3. One real life application (not in course content) of any one of the data structures given in the course syllabus.

Answer

Real life examples of stack are:

- To ***reverse a word***. You push a given word to stack - letter by letter - and then pop letters from the stack.
- An ***"undo"*** mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.
    - *Undo/Redo* stacks in Excel or Word.
- ***Language processing*** :
    - *space for parameters and local variables* is created internally using a stack.
    - compiler's *syntax check for matching braces* is implemented by using stack.
- A ***stack of plates/books*** in a cupboard.
- Wearing/Removing ***Bangles***.
- Expression evaluation and syntax parsing, many virtual machines like JVM are stack oriented.
- Support for ***recursion***
    - Activation *records of method calls.*

Reverse A Word Using Stack Process :-

1. Create an empty stack.

2. Traverse the entire string, while traversing add the characters of the string into a temporary
   variable until you get a space(‘ ‘) and push that temporary variable into the stack.

3. Repeat the above step until the end of the string.

4. Pop the words from the stack until the stack is not empty which will be in reverse order.