Biyani's Group of Girls' Colleges

I Internal Examination Sept. 2019
Class: - BCA III
Subject: - Data Structures (BCA 301)

MM: 40 Time: 1 ½ Hrs.

[I] Very short answer questions (Max 40 words). (5 * 2 = 10)

1. **What is a data structure?**
   In computer science, a **data structure** is a data organization, management and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data

2. **What are linear and non-linear data structures?**

   The data structure can be defined as the interpretation of logical relationship existing between the solitary elements of data. The linear and non-linear data structure is the subclassification of the data structure which comes under the Non-primitive data structure. The crucial difference between them is that the linear data structure arranges the data into a sequence and follow some sort of order. Whereas, the non-linear data structure does not organize the data in a sequential manner.

3. **What is stack and where it can be used?**
   In computer science, a **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.

4. **Why do we need to do algorithm analysis?**
   Algorithm analysis is an important part of a broader computational complexity theory, which provides theoretical estimates for the resources needed by any algorithm which solves a given computational **problem**. These estimates provide an insight into reasonable directions of search for efficient algorithms.

5. **What do you understand by GET and FREE node?**

The *getnode* operation may be regarded as a machine that manufactures nodes. Initially there exist a finite pool of empty nodes and it is impossible to use more than that number at a given instant . If it is desired to use more than that number over a given period of time, some nodes must be reused. The function of *freenode* is to make a node that is no longer being used in its current context available for reuse in a different context.

The list of available nodes is called the **available list.** When the available list is empty that is all nodes are currently in use and it is impossible to allocate any more, overflow occurs.

[II] Short answer questions (Max 80 words). ***(2 * 5 = 10)***

1. **What are various operations that can be performed on various data structures?**
   The basic operations that are performed on data structures are as follows:
   `Insertion:`Insertion means addition of a new data element in a data structure.
   `Deletion:` Deletion means removal of a data element from a data structure if it is found.
   `Searching:`Searching involves searching for the specified data element in a data structure.
   `Traversal:`Traversal of a data structure means processing all the data elements present in it.
   `Sorting:`Arranging data elements of a data structure in a specified order is called sorting.
   `Merging:`Combining elements of two similar data structures to form a new data structure of the same type, is called merging.

2. **How is an array different from Linked List?**
   The major **difference between Array and Linked list** regards to their structure. **Arrays** are index based data structure where each element associated with an index. ... While a **linked list**is a data structure which contains a sequence **of the** elements where each element is **linked** to its next element.
   Basically, an array is a set of similar data objects stored in sequential memory locations under a common heading or a variable name.
   While a linked list is a data structure which contains a sequence of the elements where each element is linked to its next element. There are two fields in an element of linked list. One is Data field, and other is link field, Data field contains the actual value to be stored and processed. Furthermore, the link field holds the address of the next data item in the linked list. The address used to access a particular node is known as a pointer.

[III] Long answer questions (Max 150 words). ***(2 * 10 = 20)***

1. **What is a Queue? How is it different from stack? Write algorithm to implement a queue.**

Queue is an abstract data structure, somewhat similar to Stacks. Unlike stacks, a queue is open at both its ends. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.

As we now understand that in queue, we access both ends for different reasons. The following diagram given below tries to explain queue representation as data structure −

Queue

As in stacks, a queue can also be implemented using Arrays, Linked-lists, Pointers and Structures. For the sake of simplicity, we shall implement queues using one-dimensional array.

Basic Operations

Queue operations may involve initializing or defining the queue, utilizing it, and then completely erasing it from the memory. Here we shall try to understand the basic operations associated with queues −

- **enqueue()** − add (store) an item to the queue.
- **dequeue()** − remove (access) an item from the queue.

Few more functions are required to make the above-mentioned queue operation efficient. These are −

- **peek()** − Gets the element at the front of the queue without removing it.
- **isfull()** − Checks if the queue is full.
- **isempty()** − Checks if the queue is empty.

In queue, we always dequeue (or access) data, pointed by **front** pointer and while enqueing (or storing) data in the queue we take help of **rear** pointer.

2. **Write an algorithm to insert an element with a given location in the linked list.**

```
void insertFirst(int key, int data){

   //create a link

   struct node *link = (struct node*) malloc(sizeof(struct node));

   link->key = key;

   link->data = data;


   //point it to old first node
```

```
   link->next = head;


   //point first to new first node

   head = link;

}
```

Biyani's Group of Girls' Colleges

I Internal Examination Sept. 2018
Class: - BCA III
Subject: - Data Structures (BCA 301)

MM: 40 Set: B Time: 1 ½ Hrs.

[I] Very short answer questions (Max 40 words). (5 * 2 = 10)

1. **What is Recursion? And which data structure is used to perform recursion?**
   Recursion in computer science is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem (as opposed to iteration). The approach can be applied to many types of problems, and recursion is one of the central ideas of computer science.

2. **What is an algorithm?**

   An algorithm is a step by step method of solving a problem. It is commonly used for data processing, calculation and other related computer and mathematical operations.

   An algorithm is also used to manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item.

3. **What are the criteria of algorithm analysis?**
   There are several parameters for determining the efficiency of an algorithm.
   Following are these parameters:

- First of all the algorithm must work for all the types of given inputs and must avoid conditions which will force it into entering infinite loop.
- Then there are most important aspects of 1. **Time complexity 2. Space complexity**
  - It should be kept in mind that space and time are inversely proportional in algorithmic sense.
  - Lets take a problem A in account. Now first I am solving the "A" by using more variables i.e data structures, it will substantially increase the speed of problem solving. In second case I will be using less space but this will result in more time required to solve the same problem.
  - Therefore, the best algorithm is which gives balance between the time required to solve the problem and space taken by its data. We need to find a meeting point between these to factors as both are equally important.

- It depends on the problem in hand to determine which of time and space should be sacrificed to enhance the other one.

4. **What operations can be performed on stacks?**
   In computer science, a 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.

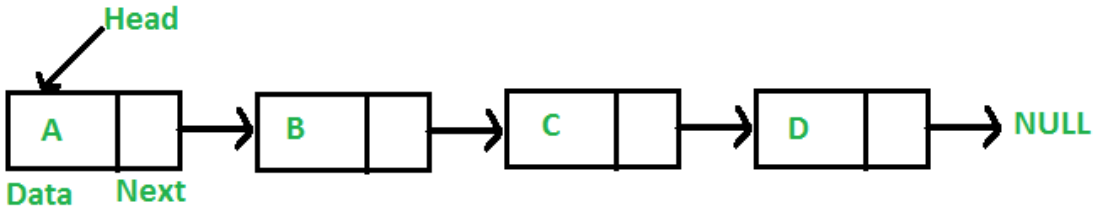
5. **What is the meaning of base address of an array?**
   Base address: A **base address** is a unique location in primary storage (or main memory) that serves as a reference point for other memory locations called absolute **addresses**. In order to obtain an absolute **address**, a specific displacement (or offset) value is added to the **base address**.

[II] Short answer questions (Max 80 words). ***(2 * 5 = 10)***

1. **What are various characteristics of an algorithm?**
   The characteristics of a good algorithm are:
   Precision – the steps are precisely stated(defined).
   Uniqueness – results of each step are uniquely definedand only depend on the input and the result of the precedingsteps.
   Finiteness – the algorithm stops after a finite number ofinstructions are executed.
   Input – the algorithm receives input.
   Output – the algorithm produces output.
   Generality – the algorithm applies to a set ofinputs.

2. **What is a linked list? Explain with example.**

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

[III] Long answer questions (Max 150 words). ***(2 * 10 = 20)***

1. **Write algorithms to perform the operations of a stack.**

```
procedure push(stk : stack, x : item):
    if stk.top = stk.maxsize:
        report overflow error
    else:
        stk.items[stk.top] ← x
        stk.top ← stk.top + 1
```

```
procedure pop(stk : stack):
    if stk.top = 0:
        report underflow error
    else:
        stk.top ← stk.top − 1
        r ← stk.items[stk.top]
        return r
```

2. **What are the various types of queues? Explain the insertion operation on a queue with algorithm.**

```
begin procedure isfull


   if rear equals to MAXSIZE

      return true

   else

      return false

   endif


end procedure
```

```
begin procedure isempty


   if front is less than MIN  OR front is greater than rear
      return true
   else
      return false
   endif


end procedure
```

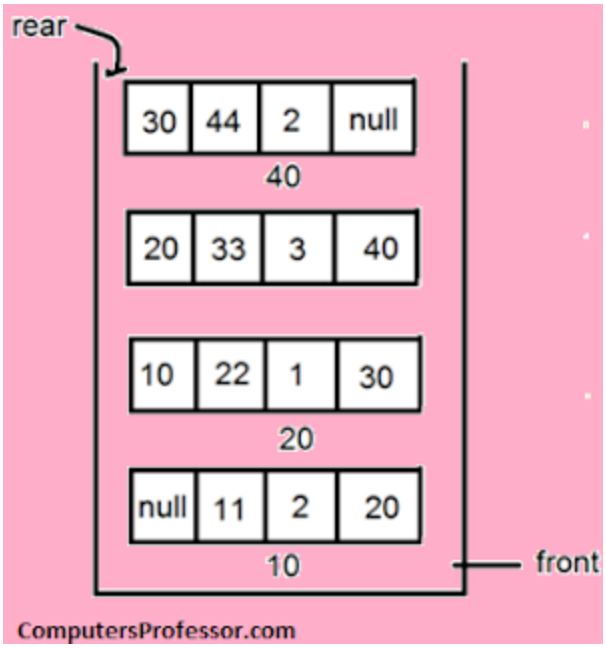
Different types of queue:

1. Abstract queue
2. Queue
3. Priority queue
4. Circular queue
5. De queue

Priority queue:

❖ In priority queues while inserting an element we assign prority to that element.
❖ While deleting we check two conditions.

1. Highest priority element deleted first.
2. While two elements have same priority the element entered first in to the queue is the deleted first then next.



Deletion order 22,11,44,33

Circular queues :

- ❖ In circular queues last element next address shows first element address (front)
- ❖ In circular queues first element previous address always shows last element address (rear).

DeQueue(double ended queue):

pronounced as deck.

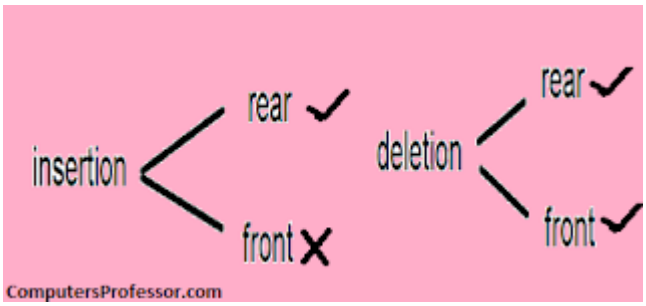
Dequeues are two types.

- ❖ Input restricted dequeue
- ❖ Out put restricted dequeue

1. **Input restricted dequeue:**

- ❖ In input restricted dequeue we restrict insertion.

- In Input restricted dequeue deletions can takes place both ends front as well as rear.
- Insertions can take place only one end called rear.



2. **Out put restricted dequeue**

- In Out put restricted dequeue we restrict deletion.
- In Out put restricted dequeue insertions can takes place both place front as well as rear.
- In Out put restricted dequeue deletion can takes place at only one end called front.