**Question 2019**

(a) **What does the data structure mean?**

A **data structure** refers to a specialized format for organizing, managing, and storing data in a way that enables efficient access and modification

Common types of data structures include:

* **Linear Data Structures**: Where elements are arranged sequentially, such as arrays, linked lists, stacks, and queues.
* **Non-Linear Data Structures**: Where elements are arranged hierarchically or more complex relationships exist, such as trees and graphs.

(b**) Write down the operations which are performed on linear data structures.**

Operations typically performed on linear data structures (like arrays, linked lists, stacks, and queues) include:

**Insertion:** Adding an element at a specific position in the structure.

**Deletion:** Removing an element from the structure.

**Traversal:** Accessing each element of the structure in sequence to process or examine it.

**Search:** Finding a specific element within the structure.

**Update:** Modifying an element in the structure.

**Sorting:** Arranging the elements in a particular order (ascending or descending).

**Merging:** Combining two or more linear data structures into a single structure.

**(c) How does an item locate in a k-D tree in an array? Explain the representation of a two-dimensional array in memory.**

2. **What is Link list? Why is linked list important in data structure?**

A **Linked List** is a linear data structure in which elements (called nodes) are not stored in contiguous memory locations. Each nodes has two parts:

1. **Data**: The actual value stored in the node.
2. **Pointer (or reference)**: A reference to the next node in the sequence.

**Importance of link list**:

 **Dynamic Memory Allocation**: Unlike arrays, linked lists allow for dynamic memory allocation.

 **Efficient Insertions and Deletions**: Linked lists provide efficient operations for insertion and deletion, particularly at the beginning and middle of the list.

 **Flexible Size**: A linked list doesn't require an initial fixed size, unlike arrays, which can allocate memory dynamically.

 **No Wasted Memory**: In arrays, extra memory may be allocated but remain unused if the array size exceeds the number of elements needed. Linked lists only use as much memory as necessary to store the elements, without wastage.

 **Efficient Memory Usage**: Since linked lists allocate memory only as needed, they can handle data sets with unpredictable sizes more efficiently than static arrays.

 **Implementation of Other Data Structures**: Linked lists serve as the building blocks for other complex data structures like stacks, queues, graphs, and hash tables.

**(b)** How to insert an item after a given node? Explain and write down the steps of algorithm.

**(c)** Mention the scenarios of header linked list and circular linked list.

### ****Header Linked List****

A **header linked list** is a variant of a linked list that contains a special node called the "header node" at the beginning of the list.

Multi-list Data Structures: Header linked lists are often used in complex data structures like **multi-lists**, where several linked lists share a common structure. The header nodes can help manage multiple lists in an organized manner.

### ****Circular Linked List****

In a **circular linked list**, the last node in the list points back to the first node, forming a circular loop instead of terminating in nullptr.

Circular Data Structures: A circular linked list is well-suited for problems where data naturally forms a loop.

**Both same scenario**

1. Easy of Insertion and Deletion
2. Efficient Traversals
3. Handling Repeating Tasks or Resources
4. Special Control Information

3.

**(a)** Define binarytree,complete binary tree and binary search tree.  
**(b)** Explain the representation of binary tree in memory. Provide traversal algorithms.  
**(c)** What are the properties of a general tree?

A close-up of a paper

Description automatically generated A close-up of a math problem

Description automatically generated



4) 4

(a) Suppose Module A requires M units of time to be executed, where M is a constant. Find the complexity C(n) of the following algorithms, where n is the input data and b is a positive integer.

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

(b) Briefly explain the following terms with respect to data structure and algorithm with proper example.

i) Recursion

**Definition:** Recursion is a technique where a function calls itself to solve a smaller subproblem of the original problem.

There are two properties

* There must be certian criteria, called base criteria which the prcedure does not call itself
* Each time the procedure does call itself, it must be closer to the base criteria.

A black screen with white text

Description automatically generated

### ii) Algorithm and procedure, ****Flowchart:****

* **Algorithm:** An algorithm is a step-by-step finite sequence of instructions that describes how to solve a problem or perform a task
* **Procedure:** A procedure is a specific implementation of an algorithm in a particular programming language. It represents a block of code or function that can be executed to perform a given task
* **flowchart** is a **graphical representation** of a process or algorithm, where each step is represented by different shapes (such as ovals, rectangles, and diamonds) connected by arrows to indicate the flow of control or execution.

**Example:**

* **Algorithm:** Sorting a list of numbers using bubble sort.
* **Procedure:** The C++ code implementing bubble sort.

2+2-4

1. a.   
   insertion sort Time complexity is  
    i) Best case:O(n)  
    ii) worst case:O(n^2)  
    iii)average case:O(n^2)

6.

**(a)** Define complete graph, neighbors, and tree graph. Draw the graph for the given adjacency matrix.

A grid of numbers with black squares

Description automatically generated  
**(b)** Explain overflow and underflow. Distinguish between linear and nonlinear data structures.

**Overflow:** sometimes new data are to be insert into a data structure but there is no available space ,this situation is usually called overflow.

It occurs when Avail=NULL

**Underflow:** refers to the situation where one wants to delete data from a data structure that is empty.

It occurs START=NULL

**(c)** Consider the following figure A2.Find a minimum path from A to J using BFS, where each edge has length 1.

A diagram of a hexagon with lines and points with Great Pyramid of Giza in the background

Description automatically generated

**Repeat exam Question 2018**

**1**

1. Suppose the following numbers are stored in an array **A**:  
   32, 51, 27, 85, 66, 23, 13, 57

Apply the bubble sort to the array A and discuss each pass separately.  
**[Marks: 07]**

**b)** Consider the linear arrays **AAA(5:50)**, **BBB(5:10)** and **CCC(18)**.

i. Find the number of elements in each array.  
ii. Suppose Base (AAA) = 300 and **w = 4** words per memory cell for **AAA**. Find the address of **AAA[15]**, **AAA[35]**, and **AAA[55]**.  
**[Marks: 07]**

**2 a**

Suppose a company keeps a linear array **YEAR(1920: 1970)** such that **YEAR[KJ]** contains the number of employees born in year **K**. Write a module for each of the following tasks:

i. To print each of the years in which no employee was born.  
ii. To find the number **NNN** of years in which no employee was born.  
iii. To find the number **N50** of employees who will be at least 50 years old at the end of the year. (Assume 1984 is the current year.)  
iv. To find the number **NL** of employees who will be at least **L** years old at the end of the year. (Assume 1984 is the current year.)  
**[Marks: 07]**

**2 b**

A hospital maintains a patient file in which each record contains the following data:  
**Name, Admission Date, Social Security Number, Room, Bed Number, Doctor Name**

i. Which items can serve as primary keys?  
ii. Which pair of items can serve as a primary key?  
iii. Which items can be group items?  
**[Marks: 07]**

**3 a**

Discuss whether a stack or a queue is the appropriate structure for determining the order in which elements are processed in each of the following situations:

i. Batch computer programs are submitted to the computer center.  
ii. Program A calls subprogram B, which calls subprogram C, and so on.  
iii. Employees have a contract which calls for a seniority system for hiring and firing.  
**[Marks: 07]**

**3 b**

Write an algorithm for Linear Search.  
**[Marks: 07]**

**4 a)** Sort the following array of elements by using radix sort algorithm.  
48, 243, 10, 423, 538, 128, 321, 543, 200

**b)** Give the advantages and disadvantages of two-way list over one-way list. Give the header linked list representation of the following polynomial equation.  
p(x,y,z)=2x8y7−5x7y3+5y2−6xz+4p(x, y, z) = 2x^8y^7 - 5x^7y^3 + 5y^2 - 6xz + 4p(x,y,z)=2x8y7−5x7y3+5y2−6xz+4

**c)** Define header linked list. Write an algorithm to find out the number of times a given item occurs in a linked list.

**5 a)** Compare BFS and DFS with examples and find out when to use which search technique.

**b)** Translate, by inspection and hand, each infix expression into its equivalent postfix expression:  
i) (A+B↑D)/(E−F)+G(A + B ↑ D) / (E - F) + G(A+B↑D)/(E−F)+G  
ii) A∗(B+D)/E−F∗(G+H/K)A \* (B + D) / E - F \* (G + H / K)A∗(B+D)/E−F∗(G+H/K)

**c)**  
i) Make a minheap from the following list of elements.  
44, 30, 50, 22, 50, 77

ii) Build a Huffman tree from the list of elements.

| **Item** | **A** | **B** | **C** | **D** | **E** | **F** |
| --- | --- | --- | --- | --- | --- | --- |
| Weight | 4 | 15 | 25 | 5 | 8 | 16 |

**6 a)** Define binary tree and 2-tree. Simulate (step by step processing) the inorder traversing mechanism of the following tree.

A screen shot of a computer screen

Description automatically generated

**b)** Write an algorithm to delete a node with a given ITEM of information.

**c)** What is garbage collection? Analyze the complexity of quick sort.

**Mid term 2021**

 **What is a linear data structure?** Demonstrate the two-dimensional array in memory. Compare the complexity of linear and binary searching algorithms. Suppose LA is a linear array with N elements and K is the positive integer such that K is less than or equal to N. Write an algorithm that inserts an ITEM into the Kth position in LA. **(8 marks)**

 **How does the computer evaluate the following infix expression:**  
10∗(12+4)−24/8?10 \* (12 + 4) - 24 / 8?10∗(12+4)−24/8? Explain it in detail.  
Suppose S consists of the following n=5n = 5n=5 letters:  
**S = A B C D E.** Find the number **C** of comparisons to sort S using quicksort. Is there any general conclusion? Define recursion with an example. Write the procedure to insert an ITEM into a queue. **(7 marks)**

**Question 2022**

**1.**

**a) Define the following terms in your own words: Data, Entity, Attributes, Records, and Data Structure. Explain algorithm complexity, a time-space tradeoff.**

**Data**: Data is the basic unit of information. Examples include a person's name, age, and gender.

**Entity**: An entity is something that has specific attributes, which may be assigned values. For example, a person is an entity with attributes such as name, age, and gender.

**Attributes**: Attributes are the characteristics or properties of an entity. For instance, a person has attributes like age, gender, and name.

**Information**: Information is meaningful or processed data.

**Field**: A field is a single, elementary unit of information representing an attribute of an entity. For the entity "Person," fields could be:

* Name: "John Doe"
* Age: "30"
* Gender: "Male"

**Record**: A record is a collection of field values associated with a given entity.

**File**: A file is a collection of entities within a given entity set.

**Algorithm complexity**: Algorithm complexity is a function f(n)f(n)f(n) that measures the time and/or space used by an algorithm in relation to the size of the input n.

**Space-time trade-off**

: Space-time trade-off refers to a choice in the solution of a data processing problem that allows one to reduce the running time of an algorithm by increasing the space used to store the data, or conversely, to reduce space requirements by allowing for longer running times.

b) **chapter book 4 array page 103**  
c) **State the steps of the binary search algorithm. What are the limitations of the binary search algorithm? Suppose the following numbers are stored in an array A:  
32,51,27,85,66,23,13,57  
You are asked to apply the bubble sort algorithm to array A and discuss each pass separately.**

linearSearch(data[],size, item)

\* set loc=0 and i=0  
\* Repeat 3 for i<size

\* if(Data[i]=item)

set loc=i+1;

return;

\* else

loc=0;

return;

\* Exit

**Binary search**

Binary(data[],seze,item)

1. set lb=0,up=size-1 and mid=(up+lb)/2;

2. Repeat 3 for lb<ub

3. if data[mid]=item

set loc=mid;

return;

else if

if(data[mid]<item)

set loc=mid+1;

else (data[mid)>item)

set loc=mid-1;

4 else

loc=0;

return;

5. Exit.

**Limitation of binary search limitation is**

* 1. The array must be sorted
  2. Sorted array which is very expensive when there are many insertions and deletions
  3. Inefficient for small array
  4. Not suitable for data structure such as tree.
  5. It does not works well in link list

Bubble sort

1. Bubble sort(data[],size)
2. Repeated 3 and 4 for (i<size)
3. repeat 4 for(j<size)
4. if (data[j]<data[j+1])

swap(data[j],sata[j+1])

Exit loop 1

exit loop 2

1. Exit.

2.

a) What is a list? Give an example of a list with several items where a few items are deleted from the list and some new items are inserted into the list. Mention the disadvantages of an array. How to recover an item using a linked list. Show the representation of the linked list in memory, including the free-storage list.

**What is a List?**

A **list** is a collection of elements arranged in a linear sequence. Each element can be accessed by its position (index).

* **Array List**: Elements are stored in contiguous সংলগ্ন memory locations.
* **Linked List**: Elements (called nodes) are stored at random memory locations, and each node points to the next one

### 2. ****Example of a List with Insertions and Deletions:****

#### Initial List:

List: [10, 20, 30, 40, 50]

এই লিষ্ট টি sorted অবস্থায় আছে

* **Delete items**: Let’s say we delete 20 and 50 from the list:

List after deletion: [10, 30, 40]

* **Insert new items**: Now, let’s insert 25 and 45 into the list:

List after insertion: [10, 25, 30, 40, 45]

**3. Disadvantages of Arrays:**

* **Fixed Size**: Arrays have a fixed size, meaning you must know the size of the array when you declare it, making it difficult to handle dynamic data.
* **Costly Insertions and Deletions**: Inserting or deleting an element in an array requires shifting other elements, which can be inefficient in terms of time complexity.
* **Memory Waste**: Arrays can waste memory if they are larger than necessary, or they can run out of space if they are too small.
* **time complexity is so high**.
* **Ineffective for large data store**

**4. How to Recover these Disadvantages Using a Linked List:**

A **linked list** overcomes these disadvantages because:

* It allows dynamic memory allocation .
* Insertion and deletion can be done in constant time (O(1)) if the pointer to the position is known.
* Memory is efficiently used, as nodes are only created when necessary.

&& **class notebook**

**Distinction Between Overflow and Underflow in a Linked List**

* **Overflow**:
  + Overflow occurs when there is no free memory left in the system to create a new node and add it to the linked list.
  + **In a linked list**, since memory is dynamically allocated, overflow usually happens when no more memory can be allocated for new nodes.
* **Underflow**:
  + Underflow occurs when an operation is attempted on an empty list, such as trying to delete a node
  + **In a linked list**, underflow happens when you try to remove a node from an empty list, or attempt to access a node that does not exist.

1. Distinguish between overflow and underflow in a linked list. Let LIST be a linked list in memory with successive nodes A and B, and node N is to be inserted in the list between A and B. Show the schematic diagram of such an insertion method. Write a procedure or algorithm to insert an ITEM after a given node and before node B. Draw a schematic diagram of the two-way list.

&& Notebook

**3.**  
a) Define and demonstrate the following terms in your own words: Binary tree, Complete binary tree, Extended binary tree, and Depth of a tree. Show the linked representation of the binary tree in memory.

### 1. ****Binary Tree****

A **binary tree** is a hierarchical data structure where each node has at most two children, referred to as the left and right child.

**Condition**: Every node can have 0, 1, or 2 children.

A black screen with a black background

Description automatically generated

### 2. ****Complete Binary Tree****

A **complete binary tree** is a binary tree where all levels are fully filled except possibly the last level, which should be filled from the left to right.

A screenshot of a computer

Description automatically generated

3.

An **Extended Binary Tree**, also known as a **Full Binary Tree**, is defined as a binary tree in which every non-leaf node has exactly two children. This means that:

1. **Every node in the tree** is either a **leaf node** (has no children) or has **exactly two children**.
2. There are **no nodes with only one child** in a full binary tree.

A screenshot of a computer

Description automatically generated

### 4. ****Depth of a Tree****

The **depth of a tree** is the number of edges from the root node to the deepest leaf node. It can also be thought of as the "height" of the tree.

#### Example:

In the following tree:

A

/ \

B C

/ \

D E

* The law is
* The depth of node A is 0 (root).
* The depth of node B and C is 1.
* The depth of node D and E is 2. The depth of the entire tree is 2.

**Link representation in memory && notebook**

b) Consider the following tree T. You are asked to simulate the preorder traversal algorithm with T and show the content of STACK at each step.

A black background with white and pink letters

Description automatically generated  
**c) Write the formal insertion procedure of heap, INSERT-HEAP (TREE, N, ITEM). Build a heap H from the numbers: 44, 30, 50, 22, 60, 55, 77, 59. Suppose a binary tree can have three cases in its nodes. How to delete an item from different cases in a binary search tree?**

### ****Case 1: Node has no children****

If the node to be deleted has no children, it is a leaf node. We can simply remove it by setting its parent's pointer to null.

#### Example:

Consider the following BST:

markdown

Copy code

50

/ \

30 70

/ \

20 40

Suppose we want to delete the node 20. Since 20 is a leaf node (no children), we simply remove it by updating the left child of 30 to null.

After deletion:

markdown

Copy code

50

/ \

30 70

\

40

### ****Case 2: Node has exactly one child****

If the node to be deleted has exactly one child, we delete the node by linking its parent directly to its child.

#### Example:

Consider the following BST:

markdown

Copy code

50

/ \

30 70

/

20

\

25

Suppose we want to delete node 20. It has one child (25). We delete 20 by making 30 point to 25.

After deletion:

markdown

Copy code

50

/ \

30 70

\

25

### ****Case 3: Node has two children****

If the node to be deleted has two children, the deletion is more complex. The node is replaced by its **inorder successor** (the smallest node in its right subtree), and the inorder successor is deleted.

#### Example:

Consider the following BST:

markdown

Copy code

50

/ \

30 70

/ \ / \

20 40 60 80

Suppose we want to delete the node 50. It has two children (30 and 70).

**Steps:**

1. Find the inorder successor of 50, which is the smallest node in the right subtree, i.e., 60.
2. Replace 50 with 60.
3. Delete 60 from its original position (as it has no left child, we use **Case 1** or **Case 2** to delete it).

After deletion:

markdown

Copy code

60

/ \

30 70

/ \ \

20 40 80

In this example, 60 takes the place of 50, and since 60 had no children, it is simply removed from the right subtree.

**4.**  
a) Mention the basis of linearity and nonlinearity of data structure with examples.  
i) Write an algorithm for inserting an element in a stack.  
b) Consider the infix expression:  
(2−(7+3)∗8)/((4−1)+5)(2 - (7 + 3) \* 8) / ((4 - 1) + 5)(2−(7+3)∗8)/((4−1)+5)  
Translate, by inspection and hand, into its equivalent postfix expression P. Evaluate the postfix expression.

**c)** Consider the following queue of characters, where QUEUE is a circular array which is allocated six memory cells: FRONT = 2, REAR = 4. QUEUE = \_\_, A, C, D, \_\_, **.  
(For notational convenience, we use "**" to denote an empty memory cell.) Describe the queue as the following operations take place:

* (a) F is added to the queue.
* (b) Two letters are deleted.
* (c) K, L, and M are added to the queue.
* (d) Two letters are deleted.
* (e) R is added to the queue.
* (f) Two letters are deleted.
* (g) S is added to the queue.
* (h) Two letters are deleted.

**5.**

a) Show the recursive solution to Towers of Hanoi problem for n=3. Also calculate the complexity with respect to your solution.

* 2.5 + 1.5

b) i) Distinguish between recursion and iteration. ii) Write the algorithm to find out Fibonacci sequence. Explain in details the working sequence of your algorithm to find out the 10th Fibonacci. iii) Sort the following array of elements by using selection/radix sort algorithm:

Copy code

504, 3, 561, 423, 538, 158, 421, 43, 66 < 10^2

* Mention the complexity of quick sort, insertion sort, radix sort and merging.
* 3 + 2 + 1

**6.**

a) i) Define finite graph, tree graph, and strongly connected graph with illustration. ii) What do you understand by Big Oh, Omega and Theta asymptotic notations?

* 2 + 2

b) Consider the following bus schedule of a bus operator company:

| **Bus No.** | **Station (Source)** | **Station (Destination)** | **Fare** | **Bus No.** | **Station (Source)** | **Station (Destination)** | **Fare** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1003 | Patuakhali | Dhaka | 650 | 3001 | Gazipur | Cox's Bazar | 1000 |
| 1006 | Dhaka | Patuakhali | 600 | 3005 | Khulna | Chattogram | 900 |
| 2007 | Barishal | Khulna | 500 | 4003 | Chattogram | Barishal | 950 |
| 2004 | Gazipur | Cox's Bazar | 700 | 4005 | Cox's Bazar | Khulna | 1200 |
| 2044 | Gazipur | Barishal | 650 | 4003 | Cox's Bazar | Barishal | 1150 |

1. Draw a labeled weighted graph considering the above bus schedule.  
   ii) Also show the linked representation of the graph.

C)

i) Find out the topological sorting of the following graph:

A diagram of a network

Description automatically generated

1. Consider the following figure, find a minimum path P from A to K using BFS where each edge has length 1

A diagram of a hexagon with lines and points

Description automatically generated

A piece of paper with text on it

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with text on it

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with text on it

Description automatically generated

A piece of paper with text

Description automatically generated

A paper with text and images

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with text and images

Description automatically generated