

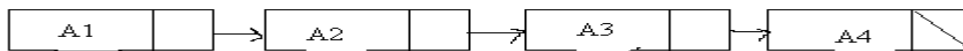
ASSIGNMENT –II
CS6202 –PROGRAMMING DATA STRUCTURE-II

1. Define Data Structures

Data Structures is defined as the way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements.

2. Define Linked Lists

Linked list consists of a series of structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a structure containing its successor. We call this theNext Pointer. The last cell'sNext pointer points to NULL.



3. State the difference between arrays and linked lists

Arrays	Linked Lists
Size of an array is fixed	Size of a list is variable
It is necessary to specify the number of elements during declaration.	It is not necessary to specify the number of elements during declaration
Insertions and deletions are somewhat difficult	Insertions and deletions are carried out easily
It occupies less memory than a linked list for the same number of elements	It occupies more memory

4. List out the advantages of using a linked list

- It is not necessary to specify the number of elements in a linked list during its declaration
- Linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list
- Insertions and deletions at any place in a list can be handled easily and efficiently
- A linked list does not waste any memory space

5. List out the disadvantages of using a linked list

- Searching a particular element in a list is difficult and time consuming
- A linked list will use more storage space than an array to store the same number of elements

6. List out the applications of a linked list

Some of the important applications of linked lists are manipulation of polynomials, sparse matrices, stacks and queues.

7. How do you initialize arrays in C? Justify an array as ADT.

```
int a[int size];
```

An array is probably the most versatile or fundamental Abstract Data Type, left until now simply to show it was reasonable to consider others. An array is a finite sequence of storage cells, for which the following operations are defined:

create(A,N) creates an array A with storage for N items;

$A[i]$ = item stores item in the i^{th} position in the array A; and
 $A[i]$ returns the value of the item stored in the i^{th} position in the array A.

8. Define an Abstract Data Type (ADT)

An abstract data type is a set of operations. ADTs are mathematical abstractions; nowhere in an ADT's definition is there any mention of how the set of operations is implemented. Objects such as lists, sets and graphs, along with their operations can be viewed as abstract data types.

9. List out the areas in which data structures are applied extensively?

The names of areas are:

Compiler Design, Operating System, Database Management System, Statistical analysis package, Numerical Analysis, Graphics, Artificial Intelligence, Simulation

10. What is radix sort?

Radix Sort is the generalised form of Bucket sort. It can be performed using buckets from 0 to 9.

In FirstPass, all the elements are sorted according to the least significant bit.

In second pass, the numbers are arranged according to the next least significant bit and so on this process is repeated until it reaches the most significant bits of all numbers.

The numbers of passes in a Radix Sort depends upon the number of digits in the numbers given.

11. Define a stack

Stack is an ordered collection of elements in which insertions and deletions are restricted to one end. The end from which elements are added and/or removed is referred to as top of the stack. Stacks are also referred as piles, push-down lists and last-in-first-out (LIFO) lists.

12. List out the basic operations that can be performed on a stack

The basic operations that can be performed on a stack are

- Push operation
- Pop operation
- Peek operation
- Empty check
- Fully occupied check

13. State the different ways of representing expressions

The different ways of representing expressions are

- Infix Notation
- Prefix Notation
- Postfix Notation

14. State the rules to be followed during infix to postfix conversions

- Fully parenthesize the expression starting from left to right. During parenthesizing, the operators having higher precedence are first parenthesized
- Move the operators one by one to their right, such that each operator replaces their corresponding right parenthesis
- The part of the expression, which has been converted into postfix is to be treated as single operand

15. State the difference between stacks and linked lists

The difference between stacks and linked lists is that insertions and deletions may occur anywhere in a linked list, but only at the top of the stack

16. Mention the advantages of representing stacks using linked lists than arrays

- It is not necessary to specify the number of elements to be stored in a stack during its declaration, since memory is allocated dynamically at run time when an element is added to the stack
- Insertions and deletions can be handled easily and efficiently
- Linked list representation of stacks can grow and shrink in size without wasting memory space, depending upon the insertion and deletion that occurs in the list
- Multiple stacks can be represented efficiently using a chain for each stack

17. Define a queue

Queue is an ordered collection of elements in which insertions are restricted to one end called the rear end and deletions are restricted to other end called the front end. Queues are also referred as First-In-First-Out (FIFO) Lists.

18. Define a priority queue

Priority queue is a collection of elements, each containing a key referred as the priority for that element. Elements can be inserted in any order (i.e., of alternating priority), but are arranged in order of their priority value in the queue. The elements are deleted from the queue in the order of their priority (i.e., the elements with the highest priority is deleted first). The elements with the same priority are given equal importance and processed accordingly.

19. State the difference between queues and linked lists

The difference between queues and linked lists is that insertions and deletions may occur anywhere in the linked list, but in queues insertions can be made only in the rear end and deletions can be made only in the front end.

20. Define a Deque

Deque (Double-Ended Queue) is another form of a queue in which insertions and deletions are made at both the front and rear ends of the queue. There are two variations of a deque, namely, input restricted deque and output restricted deque. The input restricted deque allows insertion at one end (it can be either front or rear) only. The output restricted deque allows deletion at one end (it can be either front or rear) only.

21. What is meant by sorting?

Ordering the data in an increasing or decreasing fashion according to some relationship among the data item is called sorting.

22. What are the two main classifications of sorting based on the source of data?

- a. Internal sorting
- b. External sorting

23. What is meant by external sorting?(april/may 2010)

External sorting is a process of sorting in which large blocks of data stored in storage Devices are moved to the main memory and then sorted.

24. What is meant by internal sorting?

Internal sorting is a process of sorting the data in the main memory.

25. What are the various factors to be considered in deciding a sorting algorithm?

- a. Programming time
- b. Execution time of the program

c. Memory needed for program environment

26. What is the purpose of quick sort?

The purpose of the quick sort is to move a data item in the correct direction, just enough for to reach its final place in the array.

27. How many passes are required for the elements to be sorted in insertion sort?

One of the simplest sorting algorithms is the insertion sort. Insertion sort consist of N-1 passes. For pass P=1 through N-1 , insertion sort ensures that the elements in positions 0 through P-1 are in sorted order .It makes use of the fact that elements in position 0 through P-1 are already known to be in sorted order .

28. Define Hashing.

Hashing is the transformation of string of characters into a usually shorter fixed length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the short hashed key than to find it using the original value.

29. What do you mean by hash table?

The hash table data structure is merely an array of some fixed size, containing the keys. A key is a string with an associated value. Each key is mapped into some number in the range 0 to tablesize-1 and placed in the appropriate cell.

30. What do you mean by hash function?

A hash function is a key to address transformation which acts upon a given key to compute the relative position of the key in an array. The choice of hash function should be simple and it must distribute the data evenly. A simple hash function is $\text{hash_key} = \text{key} \bmod \text{tablesize}$.

1. Explain with example the insertion & deletion in doubly linked list
2. Explain with example the insertion & deletion in singly linked list
3. .Explain with example the insertion & deletion in circularly linked list(Refer singly LL)
4. Write a C Program in polynomial addition.
5. Explain about polynomial subtraction
6. Explain Array implementation of list ADT.
7. Write a procedure to insert & delete a element in the array implementation of stack
8. Write a procedure to insert & delete a element in the linked list implementation of stack
9. Write a procedure to insert & delete a element in the array implementation of queue
10. Write a procedure to insert & delete a element in the linked list implementation of queue
11. Explain the implementation of Evaluating Postfix Expression.
12. Explain Insertion sort with example.
13. Write a note on shell sort with example.
14. Discuss briefly about heap sort with example
15. Explain Merge sort with Example.

- 16. Explain Quick sort in detail with an eg.**
- 17. Define Hash function. Write routines to find and insert an element in separate chaining.**
- 18. Explain Linear search with example?**
- 19. Explain binary search with an example?**
- 20. Explain Radix sort with an example?**