Date ________
Page ________

ASSIGNMENT OF DATA STRUCTURE

SUBMITTED TO =>

HARJEET KAUR

SUBMITTED BY =>

HARLEEN KAUR

CLASS => BCA 'A' 3RD SEM

ROLL NO => 19015161

Ques 1:- Explain time and space complexity. Also descri describe Time space trade off.

Ans 1:-

SPACE COMPLEXITY:- Space complexity of an ~~alog~~ algorithm represents the amount of memory space needed the algorithm in its life cycle.

Space needed by an algorithm is equal to the sum of the following two components.

A fixed part that is a space ~~to~~ required to store certain data and variables (i.e. Simple variables and Constants, program size etc.), that are not dependent of the size of ~~to~~ the problem.

A variable part is a space required by variables, whose size is totally dependent on the size of the problem. For example, recursion stack space, dynamic memory allocation etc.

Space complexity S(P) of any algorithm P is S(P) = A + Sp(I) where A is treated as the fixed part and S(I) is treated as the Variable part of the algorithm which depends on instance characteristic I.

TIME COMPLEXITY:- Time Complexity of an algorithm is the respresentation of the amount of time required by the algorithm to execute i ~~to~~ completion. Time requirement can be denoted or defined as a numerical function t(N), where 6 t(N) can be measued as the no. of steps, provided each step takes constant time.

For example, in case of addition of two n-bit integers, N steps are taken. ~~so~~ Consequently, the total computational tim is t(N) = c*n, where c is time consumed for addition of two bits. Here, we observe that t(N) grows linearly as input size increases.

SPACE - TIME TRADEOFF :- A space-time or time-memory trade-off in computer science is case where an algorithm or program trades increased space usage with decreased time. Here, space refers to the data storage consumed in performing a given task (computation time or respone time).

The utility of a given space-time trade off is affected by related fixe and variable cost (of, eg, CPU speed, storage space), and is subject to diminshing returns.

Ques 2:- Differentiate between Primitive and Non - Primitive data Structure.

Ans2:-

	Primitive Data Structure	Non-primitive data Structure
1.	Data structure that are directly orperated upon the machine-level instructions are known as primitive data structure.	1. The data structure that are ~~derived~~ ~~driv~~ derived from primitive data structure are called non-primitive data structure.
2.	There are basic structures are directly operated upon by the machine instructions.	2. These emphasize on structuring of a group of homogenous and heterogeneous data items.
3.	Integers, floating point, Characters, Constant, string Constants, pointers etc. fall in this category.	3. These are ~~[illegible]~~ further divided into two categories; Linear and non-linear data structure.

Ques3:- Differentiate between linear and non linear data Structure.

Ans3:-

	linear data Structure	non-linear data structure
1.	In a linear data structure, data elements are arranged in a linear order where each and every	1. In a non-linear data structure, data elements are attached in hierarchically manner.

elements are attached to its previous and next ~~ady~~ adjacent.	
2. In linear data structure, single level is involved.	2) whereas in non-linear data structure, ~~mult~~ multiple level are involved.
3. Its implementation is easy in comparison to non-linear data structure.	3) While its implementation is complex in ~~to~~ comparision to liner data Structure.
4. In linear data Structure data elements can be traversed in a single run only.	4) While in non-lianer data structure, data elements can't be traversed in a single run only.
5. In a linear data structure, memory is not utilized in an efficient way.	5. While in a non-linear data Structure memory is utilized in an efficient way.
6. Its examples are array, stack, ~~queqe~~ queue, linked list etc.	6. while its examples are : trees and graphs.
7. Application of linear data structures are mainly in application Software development.	7. Applications of non-linear data Structures are in Artifical intelligence & image processing.

Ques 4:- Suppose we have array YYY(1935:1985) in which Base(YYY)=400 and w=4 words per memory cell for YYY. find the address of YYY [1942] and YYY [1977].

Ans 4:- [LOC(LA[K]) = Base(LA) + W(K-lower bound)]

Ist case

Here,

Base (LA) = Base(YYY) = 400

w = 400

K = 1942

lower bound = 1935

Address of YYY [1942] is

LOC [YYY [1942]) = Base(YYY) + W(K-lower bound)

LOC(YYY [1942]) = 400 + 4(1942-1935)

= 400 + 4(7)

= 400 + 28

$\boxed{LOC(YYY[1942]) = 428}$

2nd case,

where K = 1977

Address of YYY[1977] is

LOC(YYY[1977]) = Base(YYY) + W(K- lower bond)

= 400 + 4 (1977-1935)

= 400 + 4(42)

= 400 + 168

$\boxed{LOC(YYY[1977]) = 568}$ Ans

Ques 5:- Differentiate between Record and file.

Ans 5:-

FILE :- A file is collection of records which have common properties. Each file has its own file reference which is unique. The file reference indicates the subject or contexts of the ~~reco~~ records.

RECORDS :- A record can be an image, text based or in electronic or physical format.