SVKM'S

Mithibai College of Arts, Chauhan Institute of Science &
Amrutben Jivanlal College of Commerce and Economics (Autonomous)
Academic Year (2022-23)
Year: 2 / Semester: IV

Program: B.Sc. Computer Science
Course: Fundamentals of Algorithms **Max. Marks: 75**
Date:

Duration: 2 ½ hrs.

REGULAR EXAMINATION

Q.1 **Attempt any three.** [21]

A What is Algorithm? Discuss different types of algorithm analysis. Which is most commonly used analysis? Why?

An **algorithm** is a step-by-step procedure for solving a problem in a finite amount of time.

1 marks

Best case, worst case, average case with notations-- **5 marks**

Worst case is used most commonly – **1 mark**

B Given following python code what is the complexity? Detail each step.

```
def func1(n):
        i=1
        while i<=n:
                i=i*2
                print(i)
        for j in range(0, n):
                print(j)
```

first loop jumps by two 1 Marks

log n execution 2 marks

second loop executes n time 1 Marks

f(n)=cn+clogn +c 2 Marks

O(n) 1 mark

C Given following python code find its complexity.

```
def func(n):
        cnt=0
        if n<=0:
                return
        for i in range(0,n) :
                for j in range(0,n):
                        cnt=cnt+1
        func(n-3)
        print(cnt)
```

f(n)=loop executes n*n time

$f(n)=n^2$ 1 Mark

$T(n)=T(n-3)+ n^2$ 2 mark

Master theorem of subtract and conquer 1 Marl

a=1

b=3

d=2

$O(n^3)$

3 Mark

D

Develop recursive python program to find x^y . Find its complexity.

```
def power(base,exp):
    if(exp==1):
        return(base)
    if(exp!=1):
        return(base*power(base,exp-1))
```

2 Marks

T(n)=T(n-1)+1   1 Mark

Master theorem of subtract and conquer   1 Mark

a=1

b=1

O(n)

3 Marks

Q.2 **Attempt any three:** [21]

A

What are heaps? Explain heapsort with following example:

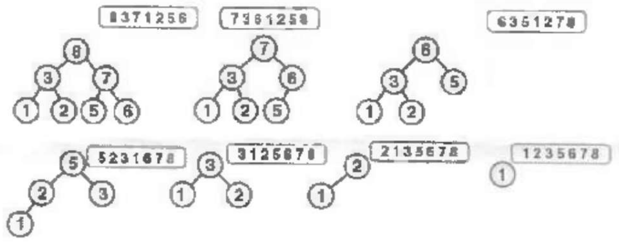
8,3,7,1,2,5,6

A heap is a complete binary tree, and the binary tree is a tree in which the node can have the utmost two children. A complete binary tree is a binary tree in which all the levels except the last level, i.e., leaf node, should be completely filled, and all the nodes should be left-justified.

2 Marks

Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to the selection sort where we first find the minimum element and place the minimum element at the beginning. Repeat the same process for the remaining elements.

2 Marks



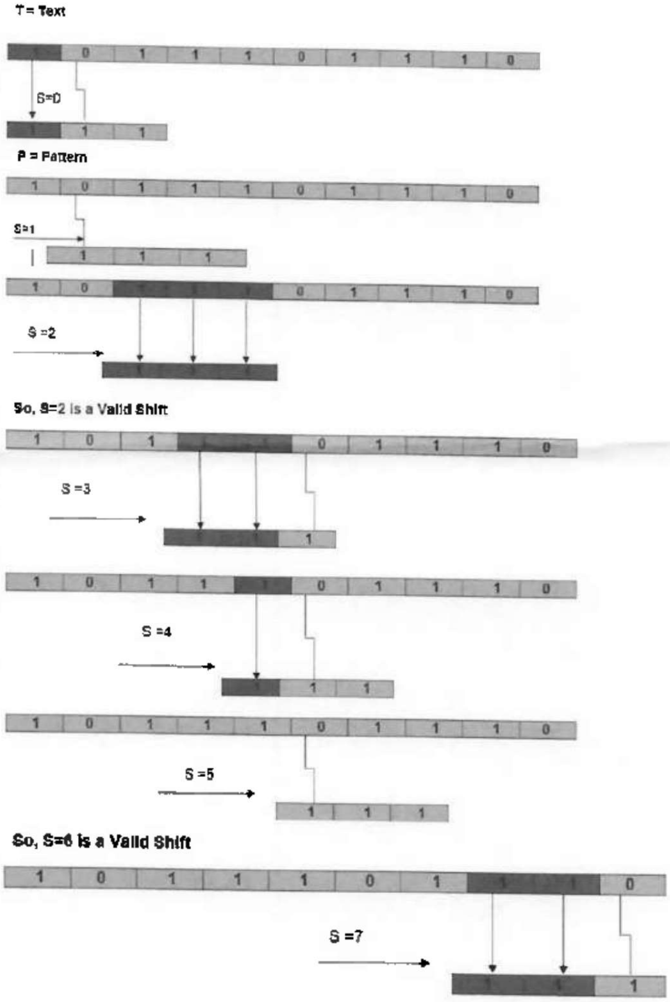
3 Marks

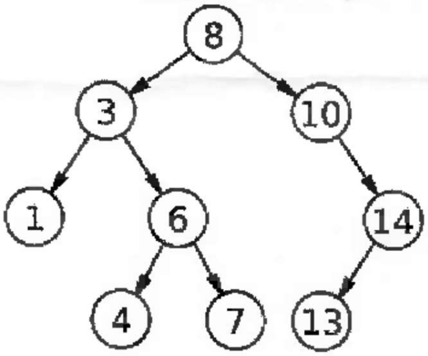
B

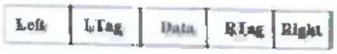
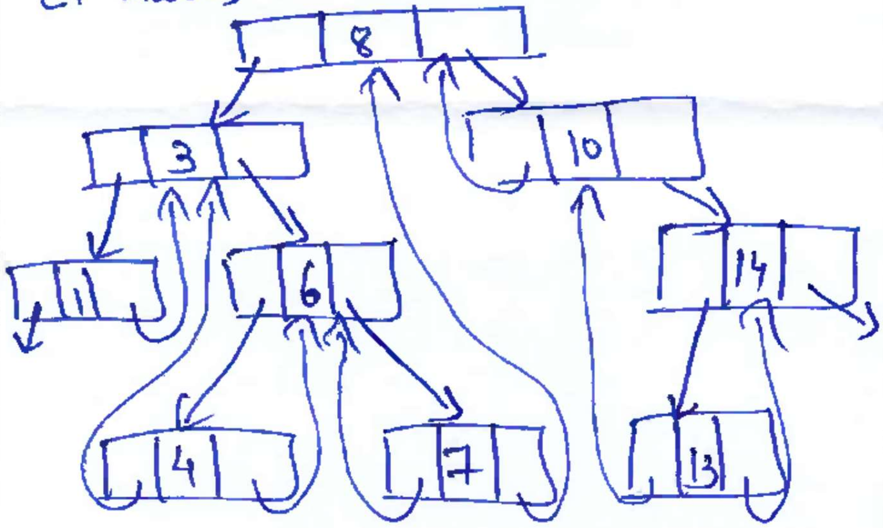
What is string matching? Describe naïve approach of the string matching with example.

The problem of finding occurrence(s) of a pattern string within another string or body of text.

The naïve approach tests all the possible placement of Pattern P [1.......m] relative to text T [1......n]. We try shift s = 0, 1.......n-m, successively and for each shift s. Compare T [s+1.......s+m] to P [1......m].

		2 Mark
		
		 T= Text
		S=0
		P = Pattern
		S=1
		S =2
		So, S=2 is a Valid Shift
		S =3
		S =4
		S =5
		So, S=6 is a Valid Shift
		S =7
		5 mark	

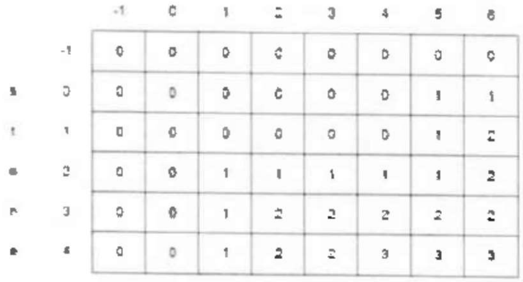
	C	Explain the concept of threaded binary tree with its node structure. Given following binary tree generate threaded binary tree.  The idea of threaded binary trees is to make inorder traversal faster and do it without stack and without recursion.	

		A binary tree is made threaded by making all right child pointers that would normally be NULL point to the inorder successor of the node 2 Marks Node Structure:  *Threaded Binary Tree Node* 2 marks Inorder traversal: 1, 3, 4, 6, 7, 8, 10, 13, 14 [1 Mark]  [2 Marks]	
	D	Discuss median of median algorithm with suitable example. • The Median of Medians is a fast recursive method for finding a value close to the median. 1. Divide the list into n/5 sublists of length 5 and perhaps one group with the remaining elements and sort those sublists. 2. Select the median of each sublist. For the final sublist if it has two or four elements select the lower median. 3. Apply selectkth recursively on the smaller list of those values in order to find the median (or lower median) of that new list. **4 Marks +3 Marks example**	
Q.3		**Attempt any three.**	[21]
	A	Explain following methods of algorithm classification: • Linear Programming ○ A method to allocate scarce resources to competing activities in an optimal manner when the problem can be expressed using a linear objective function and linear inequality constraints. 2 Mark • Reduction	

Method to solve difficult problem by transforming it into a known problem for which we have asymptotically optimal algorithms.
1 mark

- Deterministic or Non-Deterministic
  Deterministic algorithms solve the problem with a predefined process.
  1 mark
  Non-deterministic algorithm guesses the best solution at each step
  1 mark
- Exact or Approximate
  Algorithms for which we are able to find optimal solution are exact algorithms
  1 mark
  Approximate algorithms are associated with NP-hard problems where optimal solution may not be found only approximation may be possible.
  1 mark

B Find out longest common subsequence of longest and stone.



		-1	0	1	2	3	4	5	6
	-1	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	1	1
t	1	0	0	0	0	0	0	1	2
o	2	0	0	1	1	1	1	1	2
n	3	0	0	1	2	2	2	2	2
e	4	0	0	1	2	2	3	3	3

4 Marks
LCS= one
3 marks

C Write program of quick sort as an application of divide and conquer strategy.

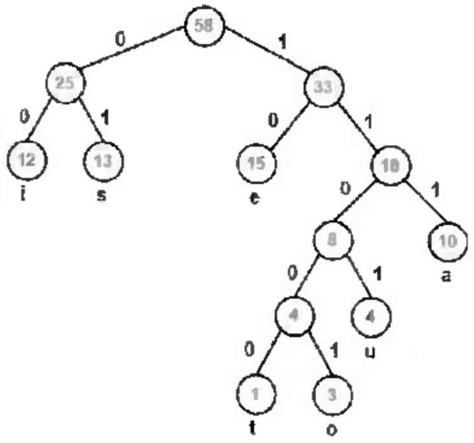
```
def quickSort(alist,first,last):
    if first<last:
      splitpoint = partition(alist,first,last)

      quickSort(alist,first,splitpoint-1)
      quickSort(alist,splitpoint+1,last)
def partition(alist,first,last):
   pivotvalue = alist[first]
   leftmark = first+1
   rightmark = last
   done = False
   while not done:

       while leftmark <= rightmark and alist[leftmark] <= pivotvalue:
            leftmark = leftmark + 1

        while alist[rightmark] >= pivotvalue and rightmark >= leftmark:
            rightmark = rightmark -1

        if rightmark < leftmark:
            done = True
        else:
            temp = alist[leftmark]
            alist[leftmark] = alist[rightmark]
            alist[rightmark] = temp

    temp = alist[first]
    alist[first] = alist[rightmark]
    alist[rightmark] = temp
    return rightmark
```

7 Marks

D Given following character frequencies:

Character	a	e	i	o	u	s	t
Frequency	10	15	12	3	4	13	1

Find its Huffman code.



Huffman Tree

5 Marks for construction of above tree

a = 111

e = 10

i = 00

o = 11001

u = 1101

s = 01

t = 11000

2 Marks for above code

Q-4 Attempt any three: [12]

A

Given recurrence relation $T(n)=16T(n/4)+n$ find its Θ.

$a=16, b=4, k=1, p=0$ 1 Mark

Master theorem of divide and conquer 1 Mark

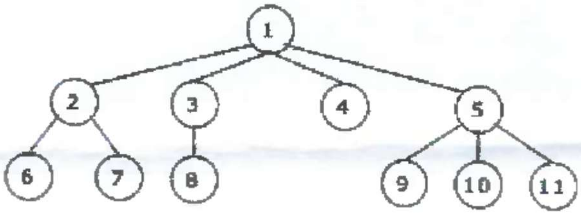
$b^k=4$

$\Theta(n^2)$

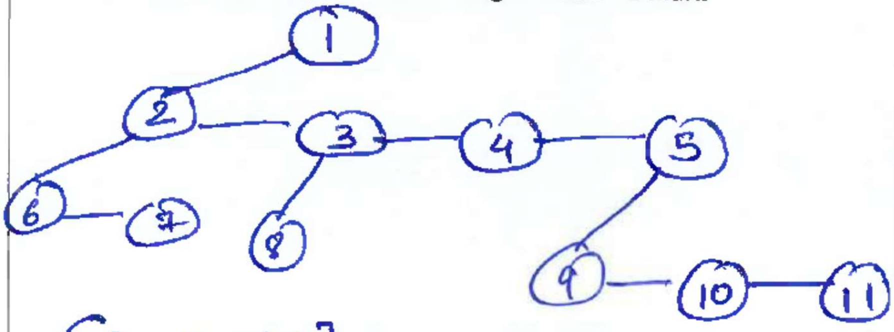
2 Marks

B

What are Generic tree? Given following generic tree convert it in corresponding binary tree.



- **Generic trees** are tree data structures that allow us to have up to n children nodes for each of the nodes **1 mark**
- For each node, we link the children of the common parent(siblings) from left to right order. **1 mark**

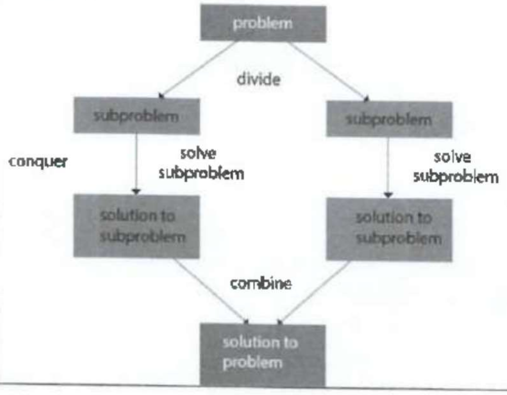


[2 marks]

C

Explain the programming terminology by which D & C divides problems in sub-problems. Draw visualization of divide and conquer strategy.

Recursion is used for divide and conquer technique.
1 mark



		3 mark	
	D	Explain components and approaches of dynamic programming Optimal substructure: optimal solution to a problem contains optimal solution to sub problems Overlapping sub problems: A recursive solution contains a small number of distinct sub problems **2 marks** Top-down (Memoization): problem is broken in sub problems and each sub problem is solved Bottom-up (Tabulation): evaluate sub problem starting with smallest possible input, store the values in table. **2 Marks**	