**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM CAMPUS**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**18CSC204J DESIGN AND ANALYSIS OF ALGORITHMS**

**CYCLE TEST-I - SET – B**

**Year: II Sem: IV Date: 18: 02:2021**

**Time: 1Hour Marks : 25**

***PART – A* (15 X 1 = 15 MARKS)**

1. \_\_\_\_\_\_ solution requires reasoning built on knowledge and experience

A. Algorithmic Solution

B. Heuristic Solution

C. Random Solution

D. Brute force Solution

Answer: - B

1. An algorithm is a \_\_\_\_\_\_\_\_\_ set of precise instructions for performing computation.

Infinite

Finite

Constant

Variable

Answer: b

1. In the principle of mathematical induction, which of the following steps is mandatory?

induction hypothesis

inductive reference

induction set assumption

minimal set representation

Answer: A

1. What does the algorithmic analysis count?

A. The number of arithmetic and the operations that are required to run the program

B.The number of lines required by the program

C.The number of seconds required by the program to execute

D.None of these

Answer:-A

1. Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

f1(n) = 2^n

f2(n) = n^(3/2)

f3(n) = nLogn

f4(n) = n^(Logn)

A. f3, f2, f1, f4

B. f2, f3, f1, f4

C. f2, f3, f4, f1

D. f3, f2, f4, f1

Answer is: D

1. The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of

n

n^2

n log n

log n

Answr C

1. The maximum gate delay for any output to appear in an array multiplier for multiplying two n bit number is

O(n^2)

O(n)

O(log n)

O(1)

Answer A

1. Which of the following statement is true?

Both time and space efficiency are measured as function of the algorithm input size

Only time efficiency are measured as function of the algorithm input size

Only space efficiency are measured as function of the algorithm input size

Neither time nor space efficiency are measured as function of the algorithm input size

Answer A

1. Consider the following segment of Code

Int j,n;

J=1;

While (j<=n)

J=j\*2;

The number of comparisons made in the execution of the loop for any n>0 is

Log n+1

N

Log+n

N Log n+1

Answer A

1. AN unordered list contains n distinct elements, The number of comparisons to find an element in the list that is neither maximum nor minimum is\_\_\_\_\_

Θ (log n)

Θ (1)

Θ (n log n)

Θ (n^2)

Answer B

1. Find the Step Count in the Given Algorithm

int sequentialSearch(••• )

{

int i;

for (i = 0; i < n && x != a[i]; i++);

if (i == n) return -1;

else return i;

}

N

N+1

N+3

N^2

Answer C

1. The data structure used to implement recursive function calls \_\_\_\_\_\_\_\_\_\_\_\_\_

Array

Linked list

Binary tree

Stack

Answer D

1. What is the base case for the following code?

void my\_recursive\_function(int n)

{

if(n == 0)

return;

printf("%d ",n);

my\_recursive\_function(n-1);

}

int main()

{

my\_recursive\_function(10);

return 0;

}

a) return

b) printf(“%d “, n)

c) if(n == 0)

d) my\_recursive\_function(n-1)

Answer: c

1. Solve the Recurrence relation T(n)=T(n/2)+1 with the initial condition T(1)=1

Θ (log n)

Θ (1)

Θ (n log n)

Θ (n^2)

Answer A

1. Solve the Recurrence relation T(n)=2T(n/2)+n with the initial condition T(1)=0

Θ (log n)

Θ (1)

Θ (n log n)

Θ (n^2)

Answer C

**PART B (2 \* 5=10 MARKS)**

**Answer any 2 Questions Shortly**

**16. Explain Analysis Framework of algorithm with Suitable Example**

The efficiency of an algorithm can be decided by measuring the performance of an algorithm. We can measure the performance by computing two factors **(2 Mark)**

1. Amount of Time required by algorithm to execute.
2. Amount of Storage required by an algorithm

***The Analysis Framework consists of (ANY 3, each carries 1 Mark)***

1. Space Complexity S(p)=C+Sp

2. Measuring an Input’s Size

3. Time Complexity – Frequency Count, System load, Instruction Set Used, No of other Programs Running

4. Units for Measuring Running Time T(n)= Cop C(n)

5. Orders of Growth – Performance of an algorithm in relation with the input size n

6. Worst-Case, Best-Case, and Average-Case Efficiencies

17. **Discuss in detail about all the asymptotic notations with Example**

Asymptotic notations are the mathematical notations used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value.

***There are mainly three asymptotic notations:***

1. Big-O notation, ii. Omega notation iii. Theta notation

**Big-O Notation (O-notation) (1 mark)**

Big-O notation represents the upper bound of the running time of an algorithm. Thus, it gives the worst-case complexity of an algorithm.

F(n)≤c\*g(n) , F(n) **∈** O (g(n))

**Omega Notation (Ω-notation) (1 mark)**

Omega notation represents the lower bound of the running time of an algorithm. Thus, it provides the best case complexity of an algorithm.

F(n)≥c\*g(n) , F(n) ∈ Ω (g(n))

**Theta Notation (Θ-notation) (1 mark)**

Theta notation encloses the function from above and below. Since it represents the upper and the lower bound of the running time of an algorithm, it is used for analysing the average-case complexity of an algorithm.

C1 \*g(n)≤F(n)≤ C2 \*g(n) F(n) ∈ Θ (g(n))

**Example (2 mark)**

18. **Explain the Recursive algorithm for Computing Tower of Hanoi and analyse its Time Complexity using Backward Substitution method**

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules: **(1 mark)**

* Only one disk can be moved at a time.
* Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
* No disk may be placed on top of a smaller disk.

***Algorithm: Tower of Hanoi* (2 mark)**

START

Procedure Hanoi(disk, source, dest, aux)

IF disk == 1, THEN

move disk from source to dest

ELSE

Hanoi(disk - 1, source, aux, dest) // Step 1

move disk from source to dest // Step 2

Hanoi(disk - 1, aux, dest, source) // Step 3

END IF

END Procedure

STOP

Recurrence Relation Equation: T(n)= 2T(n-1)+1 with initial condition T(1)=1**(2 mark)**

Time Complexity = **T(n)= Θ(2n)**

|  |  |
| --- | --- |
|  | **Verified By**  **Dr K. Raja**  **Professor & Head / CSE** |