(b) If we could always choose the median as the splitter, then we could show a linear bound on the running time. Let cn be the running time for Select(), not counting the time for the recursive call. Then, with medians as splitters, give an upper bound on the running time T(n). 2

(c) A universal family of hash functionsis a set of hash functions H mapping a universe U to the set {0,1, .., m–1} such that: 2

- For any pair of elements $u \neq v$: $Pr_{h \in H}[h(u)=h(v)] \leq 1/m$
- Can select random h efficiently and can compute h(u) efficiently.

Given U = { a, b, c, d, e, f }, m = 2, find whether $H=\{h_1,h_2\}$ is a universal family of hash functions or not in the following cases:

	a	b	c	d	e	f
$h_1(x)$	0	1	0	1	0	1
$h_2(x)$	0	0	0	1	1	1

10. Imagine you are a highly-in-demand actor, who has been presented with offers to star in n different movie projects under development. Each offer comes specified with the first and last day of filming. To take the job, you must commit to being available throughout this entire period. Thus you cannot simultaneously accept two jobs whose intervals overlap. For an artist such as yourself, the criteria for job acceptance is clear: you want to make as much money as possible. Because each of these films pays the same fee per film, this implies you seek the largest possible set of jobs (intervals) such that no two of them conflict with each other.

You (or your agent) must solve the following algorithmic scheduling problem:
Problem: Movie Scheduling Problem
Input: A set I of n intervals on the line.
Output: What is the largest subset of mutually non-overlapping intervals which can be selected from I?
You are given the job of developing a scheduling algorithm for this task.

(a) Present both optimization and decision problem formulation of the general movie scheduling problem. 2

(b) Mention some potential heuristics to be followed for developing a scheduling algorithm for this task. 2

(c) Prove that the general movie scheduling problem is NP-complete, with a reduction from independent set. 2

*** End of Questions ***

END SEMESTER EXAMINATION, JUNE-2023

ALGORITHM DESIGN-2 (CSE 4131)

Programme: B.Tech.(CSE/CSIT) **Semester: 4th**

Full Marks: 30 **Time: 2 Hours**

Course Outcome	Taxonomy Level	Ques. Nos.	Marks
CO1: understand the network flow problem and apply it to real-world problems.	L3, L4, L5	2(a), 2(b), 2(c)	2+2+2
CO2: -distinguish between computationally tractable and intractable problems - define and relate class-P, class-NP and class NP-complete, PSPACE, PSPACE-complete. - given a problem in NP, define an appropriate certificate and the verification algorithm.	L3, L4, L5,	3(a), 3(b), 3(c), 4(a), 4(b), 4(c), 5(a), 5(b), 5(c)	2+2+2+2 +2+2+2+ 2+2
CO3: understand approximation algorithms and apply this concept to solve problems	L3, L4, L5	6(a), 6(b), 6(c), 7(a), 7(b), 7(c),	2+2+2+2 +2+2
CO4: understand local search techniques and apply this concept to solve problems	L3, L4, L5,	8(a),8(b), 8(c),	2+2+2
CO5: understand randomization and apply this concept to solve problems	L3, L4, L5	9(a), 9(b), 9(c),	2+2+2
CO6: identify and apply an appropriate algorithmic approach to solve a problem and explain the challenges to solve it	L2,L3, L4,	1(a), 1(b), 1(c), 10(a), 10(b), 10(c),	2+2+2+2 +2+2

*Bloom's taxonomy levels: Knowledge (L1), Comprehension (L2), Application (L3), Analysis (L4), Evaluation (L5), Creation (L6)

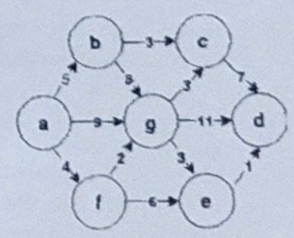
Answer all questions. Each question carries equal mark.

1\. (a) Is there a subset of the numbers in {23, 59, 17, 47, 14, 40, 22, 8} that sums to 100? What about 130? Show your work. 2

(b) The following brief list of steps will provide you with "the Right Stuff" to be an algorithm designer. 2
Step1: Do you really understand the problem?
Step2: Can you find a simple algorithm or heuristic for my problem?
Step3: Is your problem in the catalog of algorithmic problems covered in CSE3131 and CSE4131?
Step4: Are there special cases of the problem that you know how to solve?
Step5: Which of the standard algorithm design paradigms are most relevant to your problem?
Are you still stumped? Then in which of the above step(s), you can apply some extra efforts in terms of some substeps?

(c) Find the most suitable match between Gr-A (P, PSPACE, NPC, NPH) and Gr-B (Halting problem, MST, QSAT, TSP).2

2\. (a) Considering vertices a and d as source and sink vertices, find the maximum flow from a to d. 2



(b) Find the minimum number of iterations to find the answer in Q2(a). 2

(c) Show the minimum cut and the corresponding cut-edges. 2

3. (a) Draw an example of a graph with 10 vertices and 15 edges that has a vertex cover of size 2. 2

(b) Reduce a 3-SAT formula to an independent set problem. (Explain with an example). 2

(c) What are the basic features of a computationally hard problem that can be sacrificed for dealing with intractable problems? 2

4. (a) Given an instance of 8-puzzle game as follows. 2

1	2	3
4	5	8
7	-	6

Map the problem into an instance of planning problem.

(b) Identify each operators and their corresponding prerequisite list, add list and delete list in the planning model designed in Q4(a). 2

(c) Given an instance of the QSAT problem as follows. 2

$\phi(x1, x2, x3) = (x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3)$

Draw the recursion tree that leads to all possible truth assignments for the given QSAT. Check each assignment for the solution of ϕ.

5. (a) Consider a statement - "The intractability of Vertex Cover decision problem only sets in for real once k grows as a function of n." (where n, the number of nodes in the graph, and k, the allowable size of a vertex cover.) Justify the statement with proper reasoning. 2

(b) Differentiate between tractable and intractable problem. 2

(c) Give a precise definition of a NP-Complete problem. How can we prove a given problem as a NP-Complete problem? 2

6.
```
Greedy-Balance (n,m,t[1...n]) {
    Start with no jobs assigned
    Set Ti = 0 and A(i) = ∅ for all machines Mi
    For j = 1, . . . , n do {
        Let Mi be a machine that achieves the minimum min_k T_k
        Assign job j to machine Mi
        Set A(i) ← A(i) ∪ {j}
        Set Ti ← Ti + tj
    }
    return A[1],A[2],...,A[m]
}
```

(a) Show that algorithm Greedy-Balance produces an assignment of jobs to machines with makespan $T \le 2T^*$. 2

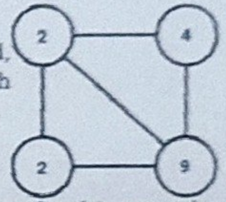
(b) What will be the resulting makespan of running this greedy algorithm on a sequence of six jobs with processing times 2, 3, 4, 6, 2, 2 for m=3 indentical machines? 2

(c) In the laod balancing problem, suppose we have m machines and $n = m(m - 1) + 1$ jobs. The first $m(m - 1) = n - 1$ jobs each require time $t_j = 1$. The last job is much larger; it requires time $t_n = m$. What does our above greedy approximation algorithm do with this sequence of jobs? Show that the approximation ratio is close to a factor of 2 when m is large. 2

7. (a) Prove that: "For any vertex cover S, and any nonnegative and fair prices p_e, we have $\Sigma_{e \in E}\, p_e \le w(S)$." (Fairness lemma) 2

(b) Given an approximation algorithm for weighted-vertex-cover problem as below: 2

```
Vertex-Cover-Approx( G, w ) {
    Set pe = 0 for all e ∈ E
    While (there is an edge e = (i, j) such that neither i nor j is tight) {
        Select such an edge e
        Increase pe without violating fairness
    }
    Let S be the set of all tight nodes
    Return S
}
```



Using the above algorithm based on pricing method, find the weighted-vertex-cover for the given graph with four vertices having vertex weights/costs 2, 2, 4 and 9.

(c) Compare your solution in Q7(b) with the optimal solution for this example. 2

8. Given an undirected interger-weighted graph $G=(V_s,E_w)$ that represents a Hopfield Neural Network where set V_s represents the set of all vertices with their states as pair of (vertex,state) and E_w represents the set of all edges with their edge-weights as pair of (edge,edge-weight) which is as follows:

$V_s = \{(v_1,-1),(v_2,+1),(v_3,+1),(v_4,+1),(v_5,-1)\}$ and

$E_w = \{((v_1,v_2),-10),((v_1,v_3),8),((v_2,v_3),-4),((v_4,v_3),-1),((v_5,v_3),-1)\}$

(a) For the above given configuration of the Hopfield neural network, find the sum of weights of all good edges. 2

(b) Using the State-flipping algorithm, show all the steps to find a stable configuration from the given configuration of the Hopfield neural network. 2

(c) Give a suitable bound for the number of iterations for the State-flipping algorithm to find a stable configuration. 2

9.
```
Select(S, k)
{choose a splitter ai ∈ S, uniformly at random
  for each element aj of S {
        if aj < ai then put aj in S⁻
        if aj > ai then put aj in S⁺
  }
  if |S⁻| = k − 1 then
        return ai      // the splitter ai was in fact the desired answer
  else if |S⁻| ≥ k then
        Select(S⁻,k)  // the kth largest element lies in S⁻
  else // suppose |S⁻| = l < k −1
        Select(S⁺,k)   // the kth largest element lies in S⁺
}
```

(a) "Regardless of how the splitter is chosen, the algorithm Select(S,k) above returns the k^{th} largest element of S." – TRUE/FALSE. Justify your answer. 2