CSE 4131: ALGORITHM DESIGN II

ASSIGNMENT 2:

Submission due date: 14th June 2023

- *Assignment scores/markings depend on neatness and clarity.*
- *Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.*
- *The marking would be out of 100.*
- *You are allowed to use only those concepts which are covered in the lecture class till date.*
- *Plagiarized assignments will be given a zero mark.*

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.

CO3: understand approximation algorithms and apply this concept to solve some problems for which polynomial time exact solutions are probably unattainable.

CO6: identify and apply an appropriate algorithmic approach to solve a problem and explain the challenges to solve it.

<table>
<tr><th>Sl.No.</th><th>Question</th><th>PO</th><th>Level</th></tr>
<tr><td>1.</td><td>Using the (conditions, operators) model, formulate a planning instance for the fifteen-puzzle problem — a 4 × 4 grid with fifteen movable tiles labeled as 1, 2, . . . , 15, and a single hole, with the goal of moving the tiles around so that the numbers on the tiles end up in ascending order.</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>2.</td><td>Given an instance of 8-puzzle game as follows.
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>8</td></tr>
<tr><td>7</td><td></td><td>6</td></tr>
</table>
a. Map the problem into an instance of planning problem.
b. Identify each operators and their corresponding prerequisite list, add list and delete list.
c. Can we achieve the goal state from the current configuration? Justify your answer.</td><td>PO1, PO2</td><td>L2, L3, L4</td></tr>
<tr><td>3.</td><td>Prove the following statement. “If a Planning instance with n conditions has a solution, then it has one using at most $2^n - 1$ steps.”</td><td>PO1, PO2</td><td>L2, L3, L4</td></tr>
</table>

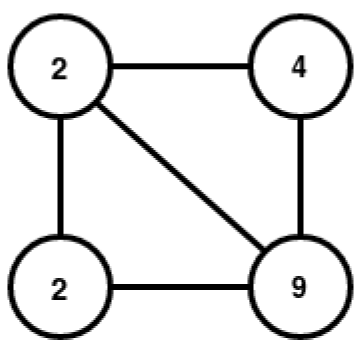
4.	Given an instance of the QSAT problem as follows. $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4} \vee \overline{x_5}) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_5) \wedge (\overline{x_2} \vee x_4 \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_5) \wedge (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_3 \vee \overline{x_4} \vee x_5)$ $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \forall x_5\ \varphi\ (x_1, x_2, x_3, x_4, x_5)$? Draw the recursion tree that leads to all possible truth assignments for the given QSAT. Check each assignment for the solution of φ.	PO1, PO3	L2, L3, L4
5.	Given a new problem P, being a student of Algorithm Design, discuss in detail, the possible options/approaches you will explore to solve the problem using a computer.	PO1, PO4	L2, L3, L4
6.	Prove that, COMPETITIVE-FACILITY-LOCATION ∈ PSPACE-complete.	PO1, PO2	L2, L3, L4
7.	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 is the number of nodes in the graph, and k is the allowable size of a vertex cover.) Justify the statement with proper reasoning.	PO1, PO2	L2, L3, L4
8.	The Uncapacitated facility location problem can be defined as follows. Given a set D of clients and a set F of potential facility locations, a distance function d : D × F → R^+ is defined which denotes a cost d_{ij} of assigning client *j* to facility *i*, and a cost function f : F → R^+ associated with each facility $i \in F$. The goal is to choose a subset of facilities S ⊆ F so as to minimize the total cost of the facilities in S and the cost of assigning each client j ∈ D to the nearest facility in S. i.e., **Output:** S ⊆ F that minimizes $\sum_{i \in S} f_i + \sum_{j \in D} (\min_{i \in S} d_{ij})$ where f_i: cost of facility *i* and d_{ij} is the distance function value between client *i* and facility *j*. a. Give an $O(\log\lvert D\rvert)$ approximation algorithm for the given Uncapacitated facility location problem. b. Prove the bound on the approximation ratio.	PO1, PO2, PO3	L2, L3, L4
9.	Greedy-Balance (*n*, *m*, *t*[1...*n*]) { Start with no jobs assigned Set $T_i = 0$ and *A*(*i*) = ∅ for all machines M_i	PO1, PO2, PO3	L1, L2, L3, L4

	For j = 1, . . . , n do {
	Let M_i be a machine that achieves the minimum load $\min_k T_k$
	Assign job j to machine M_i
	Set $A(i) \leftarrow A(i) \cup \{ j \}$
	Set $T_i \leftarrow T_i + t_j$
	}
	return $A[1], A[2], \ldots, A[m]$
	}
	(a) Show that the algorithm Greedy-Balance produces an assignment of jobs to machines with a makespan T ≤ $2T^*$.
	(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 three identical machines?
	(c) In the load balancing problem, suppose we have m machines and $n = m(m - 1) + 1$ jobs. Each of the first $m(m - 1) = n - 1$ jobs require time $t_j = 1$. The last job is much larger with processing 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.		

10.	Greedy-Center-Selection(S, d, k) Assume that $k \leq \|S\|$ (If not then return) Select any site s ∈ S and let $C \leftarrow \{s\}$ While $\|C\| < k$ Select a site s ∈ S that maximizes $d(s, C)$ Add s to C: $C \leftarrow C \cup \{s\}$ End while return C as the set of selected centers Show that Greedy-Center-Selection algorithm returns a set C of k centers such that the covering radius $r(C) \leq 2r(C^*)$, where C^* is an optimal set of k centers.	PO1, PO2	L2, L3, L4
11.	Consider an example scenario of Center-Selection problem with only two sites s and z, and the number of centers k = 2. Assume that s and z are located in the plane, with distance equal to the standard Euclidean distance in the plane, and that any point in the plane is an option for placing a center. Let d be the distance between s	PO1, PO2	L2, L3, L4

	and z. Then find the best location for (i) **Case1:** a single center c_1 and (ii) **Case2:** then subsequent best location for second center c_2. In each case, also find the optimal covering radius $r(C)$.		

12.	Given an approximation algorithm for weighted-set-cover problem as below:
	Greedy-Set-Cover ($U, S_1, \dots, S_m, w$) {
	Start with $R \leftarrow U$ (No element is covered yet)
	Initialize $C \leftarrow \emptyset$ (No set is selected yet)
	While $R \neq \emptyset$
	Select a set S_i that minimizes $\frac{w_i}{\lvert S_i \cap R \rvert}$
	Add S_i to C: $C \leftarrow C \cup \{S_i\}$
	Delete the elements of S_i from R
	End while
	Return C as the set cover
	
	Using the above algorithm, find the solution C for the following set-cover problem instance and compare this solution with the optimal solution C^*.
	A Set-Cover instance is given as, U={1,2,3,4,5,6,7,8}, and list of subsets S1={1,3,5,7}, S2={2,4,6,8}, S3={1}, S4={2}, S5={3,4}, S6={5,6,7,8} and the weight array $w[\ ] = \{1+\varepsilon, 1+\varepsilon, 1,1,1,1,1,1\}$, where ε is a very small value between 0 and 1.
		PO1, PO2, PO3	L2, L3, L4

13.	Given an approximation algorithm for weighted-vertex-cover problem as below:
	Vertex-Cover-Approx(G, w) {
	Set $p_e = 0$ for all $e \in E$
	While (there is an edge $e = (i, j)$ such that neither i nor j is tight) {
	Select such an edge e
	Increase p_e without violating fairness
	}
	Let S be the set of all tight nodes
	Return S
	}
	
	
	Using the above algorithm, find the weighted-vertex-
		PO1, PO2, PO3	

	cover for the given graph with four vertices having vertex weights/costs 2, 2, 4 and 9. Compare your solution with the optimal solution for this example.		
14.	Prove that: "For any vertex cover S, and any nonnegative and fair prices p_e, we have $\Sigma_{e \in E}\, p_e \leq w(S)$." (Fairness lemma)	PO1, PO2	L2, L3
15.	Show that "the set S returned by the algorithm (mentioned in Q13) is a vertex cover, and its cost is at most twice the minimum cost of any vertex cover."	PO1, PO2	L2, L3
16.	Give a critical comparison between the two versions of the Dynamic Programming based solutions described as follows to solve the Knapsack problem. i. The first version formulate the structure of the sub problems as OPT(i, w): the sub problem to compute the maximum value of any solution using a subset of items 1,…,i and a knapsack of capacity w. ii. The second version formulate the structure of the sub problems as OPT(i, v): the sub problem of finding the minimum weight of a knapsack for which we can obtain a solution of value at least v using a subset of items 1,…,i.	PO1, PO2	L2, L3
17.	Give a comparison between Approximation algorithm, Approximation scheme and Fully polynomial-time approximation scheme with examples.	PO1, PO2	L2, L3
18.	Though Vertex-Cover $\leq_p$ Set-Cover, why do they have different lower bounds on the approximability?	PO1, PO2	L2, L3
19.	Compare the fairness conditions used in the approximation solutions for Weighted-Set-Cover and Weighted-Vertex-Cover problems.	PO1, PO2	L2, L3
20.	Show that the approximation algorithm we studied in the class for the Knapsack problem always runs in polynomial time even if the values and weights are large and achieves the approximation ratio (1+ε).	PO1, PO2	L2, L3

Submission and Grading:

Submit the hard copy of your assignment by the due date, i.e. 14.06.2023.

Part of your assignment grade comes from its "external correctness." This is based on correct output on various sample inputs.

The rest of your assignment's score comes from "internal correctness." Internal correctness includes:
1. Use of methods to minimize the number of steps.
2. Appropriate use of rules, axioms, and suitable diagrams to enhance readability of your responses.

Send a zip folder (name of the zip folder must be your registration number_AD2) containing the code and output file/screen-shot of each program implementation mentioned to the official email id of your AD2 class teacher. On the top of each program, you must mention your full name, registration number, title of the program and date.