CECS 528, In Class Assignment 4, February 13th, 2026, Dr. Ebert

Directions:

1. Form a group with two other students to solve each of the following problems.
2. Each problem is to be solved collectively on a single sheet of paper with students taking turns completing each stage of the algorithm and initializing the work they did for that stage.
3. All solutions should be turned in before leaving class. Make sure your names are on each sheet.

Problems

1. Recall the use of the disjoint-set data structure for the purpose of improving the running time of the `Unit Task Scheduling` algorithm. For each task below,

Task	a	b	c	d	e	f
Deadline Index	3	4	5	5	5	3
Profit	60	50	40	30	20	10

   and on a single sheet of shared paper, show the current schedule and M-Tree forest after it has been scheduled (or at least has attempted to be scheduled in case there is no time for it). Note that the earliest possible deadline index is 1, meaning that the earliest slot in the schedule array has index 1. Students should take turns processing a task and drawing the updated schedule and M-Tree forest. Since there are six tasks, it means that each student will have two turns. Please initialize your work for each of your turns. To receive full credit, your solution should show six different updates. See Example 2.13 of the Greedy Algorithms Introduction (Annotated) lecture for more guidance. (12 pts)

2. For the weighted graph with edges

$$(a, e, 6), (b, e, 4), (c, e, 3), (c, f, 5), (d, f, 2), (e, f, 1),$$

   and on a single sheet of shared paper, show how the disjoint-set data structure forest changes when processing each edge in Kruskal's sorted list of edges. Students should take turns processing a single edge, redrawing the entire M-Tree forest, and initializing their work. Since there are six edges to process, each student will take two turns. See Example 2.5 of the Greedy Graph Algorithms (Annotated) lecture for more guidance. (12 pts)

3. An instance of the `Set Cover` decision problem is a pair $(\mathcal{S}, m)$, where $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of n subsets, where $S_i \subseteq \{1, \ldots, m\}$, for each $i = 1, \ldots, n$. The problem is to find a (preferably) small subset $\mathcal{C} = \{B_1, \ldots, B_k\}$ of $\mathcal{S}$ for which

$$B_1 \cup B_2 \cup \cdots \cup B_k = \{1, 2, \ldots, m\}.$$

   In this case we say that $\mathcal{C}$ **covers** $\{1, \ldots, m\}$. Each student should take turns selecting a set to add to the cover $\mathcal{C}$. The greedy choice of set should be the one that possesses the most number of **uncovered** elements of $\{1, \ldots, m\}$, where element j is uncovered iff there is currently no subset $B \in \mathcal{C}$ for which $j \in B$. Assume $m = 20$ and

$$\mathcal{S} = \{\{5, 14, 18\}.\{1, 2, 16\}, \{2, 5, 8, 11, 13\}, \{3, 8, 13, 15, 17, 19\}, \{2, 4, 5, 15\},$$

$$\{10, 15, 18\}, \{10, 14, 15, 16, 19, 20\}, \{1, 6, 9, 14, 16\}, \{5, 6, 7, 8, 20\}, \{1, 2, 9, 12, 16, 17\}\}.$$

Break any ties by giving precedence to the first set to appear in $\mathcal{S}$ (from left to right) that has the most uncovered elements. (12 pts)

4. **Bonus Problem.** Design an instance $(\mathcal{S}, m = 2^k)$ of `Set Cover` where $k \geq 1$, the optimal cover requires just two sets, but the greedy algorithm forms a cover with at least k subsets.