

## Problem

LO3. Solve each of the following problems.

- (a) Recall the use of the disjoint-set data structure for the purpose of improving the running time of the **Unit Task Scheduling** (UTS) algorithm. For the set of tasks

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

show the M-Tree forest after it has been inserted (or at least has attempted to be inserted in case the scheduling array is full). Notice that the earliest deadline is 1, meaning that the earliest slot in the schedule array has index 1. Hint: to receive credit, your solution should show six different snapshots of the M-Tree forest.

- (b) Demonstrate the greedy algorithm for **Unit-Task Scheduling** (UTS) for the UTS instance shown below. Make sure to show the sorted order.

<b>Task</b>	a	b	c	d	e	f	g	h	i	j	k	l
<b>Deadline</b>	5	4	6	2	6	8	4	2	2	7	3	5
<b>Profit</b>	40	80	40	90	70	10	80	80	70	90	90	10

LO4. Answer the following.

- (a) The dynamic-programming algorithm that solves the **Edit Distance** optimization problem defines a recurrence for the function  $d(i, j)$ . In words, what does  $d(i, j)$  equal? Hint: do *not* write the recurrence (see Part b).
- (b) Provide the dynamic-programming recurrence for  $d(i, j)$ .
- (c) Apply the recurrence from Part b to the words  $u = \text{baccbb}$  and  $v = \text{abbbcc}$ . Show the matrix of subproblem solutions and use it to provide an optimal solution.

LO5. Consider the 2SAT instance

$$\mathcal{C} = \{(x_1, \bar{x}_2), (x_1, x_5), (x_1, x_2), (\bar{x}_1, x_4), (\bar{x}_1, x_6), (\bar{x}_2, \bar{x}_5), (x_3, x_4), (\bar{x}_3, \bar{x}_6), (\bar{x}_4, \bar{x}_5)\}.$$

- (a) Draw the implication graph  $G_{\mathcal{C}}$ .
- (b) Perform the **Improved 2SAT** algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced 2SAT instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for  $\mathcal{C}$  or indicate why  $\mathcal{C}$  is unsatisfiable.
- (c) Suppose 2SAT instance  $\mathcal{C}$  has three variables and, when running the original 2SAT algorithm the answer to each oracle query is shown in the table below. Is  $\mathcal{C}$  satisfiable? If yes, provide a satisfying assignment for  $\mathcal{C}$ . If not, explain why.

Oracle Query	Answer
$\text{reachable}(G_C, x_1, \bar{x}_1)$	Yes
$\text{reachable}(G_C, \bar{x}_1, x_1)$	No
$\text{reachable}(G_C, x_2, \bar{x}_2)$	Yes
$\text{reachable}(G_C, \bar{x}_2, x_2)$	Yes
$\text{reachable}(G_C, x_3, \bar{x}_3)$	No
$\text{reachable}(G_C, \bar{x}_3, x_3)$	Yes

LO6. Answer the following.

- Provide the definition of what it means to be a mapping reduction from problem  $A$  to problem  $B$ .
- Given **Subset Sum** instance  $(S = \{4, 6, 7, 16, 21, 25, 30, 37, 42\}, t = 96)$ , provide  $f(S, t)$ , where  $f : \text{SS} \rightarrow \text{SP}$  is the mapping reduction from **SS** to **Set Partition** described in lecture.
- Verify that  $f$  is valid for input  $(S, t)$  from part b in the sense that both  $(S, t)$  and  $f(S, t)$  are positive instances of their respective problems. Make sure to provide the solutions for both problem instances.

LO7. An instance of the **Odd Cycle** decision problem is a simple graph  $G = (V, E)$ . The problem is to decide if  $G$  has a an **odd cycle**, i.e. a sequence of  $k+1$  vertices for which i)  $k \geq 3$  is odd, ii) the first and last vertices are identical, iii) each vertex is adjacent to the vertex that immediately follows it, and iv) with the exception of the first/last vertex, no other vertex appears more than once. For example, the graph shown below has a 3-cycle via the sequence  $C = c, d, e, c$ .

- For a given instance  $G$  of **Odd Cycle**, describe a certificate in relation to  $G$ .
- Provide a semi-formal verifier algorithm that takes as input i) an instance  $G$  of **Odd Cycle**, ii) a certificate for  $G$  as defined in part a, and decides if the certificate is valid for  $G$ .
- Provide the two size parameters for **Odd Cycle**.
- Use the size paramters from part c to provide a big-O bound on the verifier's running time. Justify your answer.

