CECS 528 LO5 Assessment 3/5/2025, Dr. Ebert

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 6T(n/2) + n^2 \log^2 n$. Defend your answer. Note: for cases 1 and 3 of the Master Theorem, you must provide an appropriate use of an ε in your analysis.

(b) Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 4T(n/2) + 5n^{3/2},$$

then $T(n) = O(n^2)$.

Assume $T(k) \le ck^2 + dk^{3/2}$ for all $k<n$ and some const. $c>0$ and const. d.

Shows $T(n) \le cn^2 + dn^{3/2}$

$$T(n) = 4T(n/2) + 5n^{3/2} \le 4c\left(\frac{n}{2}\right)^2 + d\left(\frac{n}{2}\right)^{3/2} + 5n^{3/2}$$

$$= cn^2 + \frac{dn^{3/2}}{2^{3/2}} + 5n^{3/2} \le cn^2 + dn^{3/2} \Longleftrightarrow$$

$$d\left(1 + \frac{1}{2^{3/2}}\right) \ge 5 \Longleftrightarrow d \ge \frac{5}{\left(1 + \frac{1}{2^{3/2}}\right)}$$

LO2. Solve the following problems.

(a) Provide the divide-and-conquer recurrence that governs the number of steps required by the algorithm (as described in the exercises) that solves the `Minimum Positive Subsequence Sum (MPSS)` problem. Defend each part of your provided answer.

$T(n) = 2T(n/2) + n\log n$

Two subproblems, both of size $n/2$. $n \log n$ steps required to create and sort left and right sums and scan them inorder to compute the mpss.

(b) Given the matrices

$$A = \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{pmatrix} 1 & -3 \\ -4 & 5 \end{pmatrix} \\ \begin{array}{cc} c & d \end{array} \end{array} \quad B = \begin{array}{c} \begin{array}{cc} e & f \end{array} \\ \begin{pmatrix} 3 & -1 \\ 2 & 4 \end{pmatrix} \\ \begin{array}{cc} g & h \end{array} \end{array}$$

Compute AB in two different ways: i) in the standard way (using dot products), and ii) using Strassen's algorithm. In the latter case, show the calculation for each product $P_1, \ldots, P_7$. Hint: $r = ae + bg$, $s = af + bh$, $t = ce + dg$, and $u = cf + dh$ are the four entries of AB, and Strassen's products are obtained as follows.

$$A_1 = a, B_1 = f - h, A_2 = a + b, B_2 = h, A_3 = c + d, B_3 = e, A_4 = d, B_4 = g - e,$$

$$A_5 = a + d, B_5 = e + h, A_6 = b - d, B_6 = g + h, A_7 = a - c, B_7 = e + f.$$

$P_1 = (1)(-5) = -5$ $\quad P_2 = (-2)(4) = -8$ $\quad P_3 = (1)(3) = 3$ $\quad P_4 = (5)(-1) = -5$

$P_5 = (6)(7) = 42$ $\quad P_6 = (-8)(6) = -48$ $\quad P_7 = (5)(2) = 10$

$r = ae + bg = \boxed{-3} = P_5 + P_6 - P_2 + P_4 = 42 + (-48) + 8 - 5 = \boxed{-3}$ ✓

$s = af + bh = \boxed{-13} = P_1 + P_2 = -5 + (-8) = \boxed{-13}$ ✓

$t = ce + dg = \boxed{-2} = P_3 + P_4 = 3 + (-5) = \boxed{-2}$ ✓

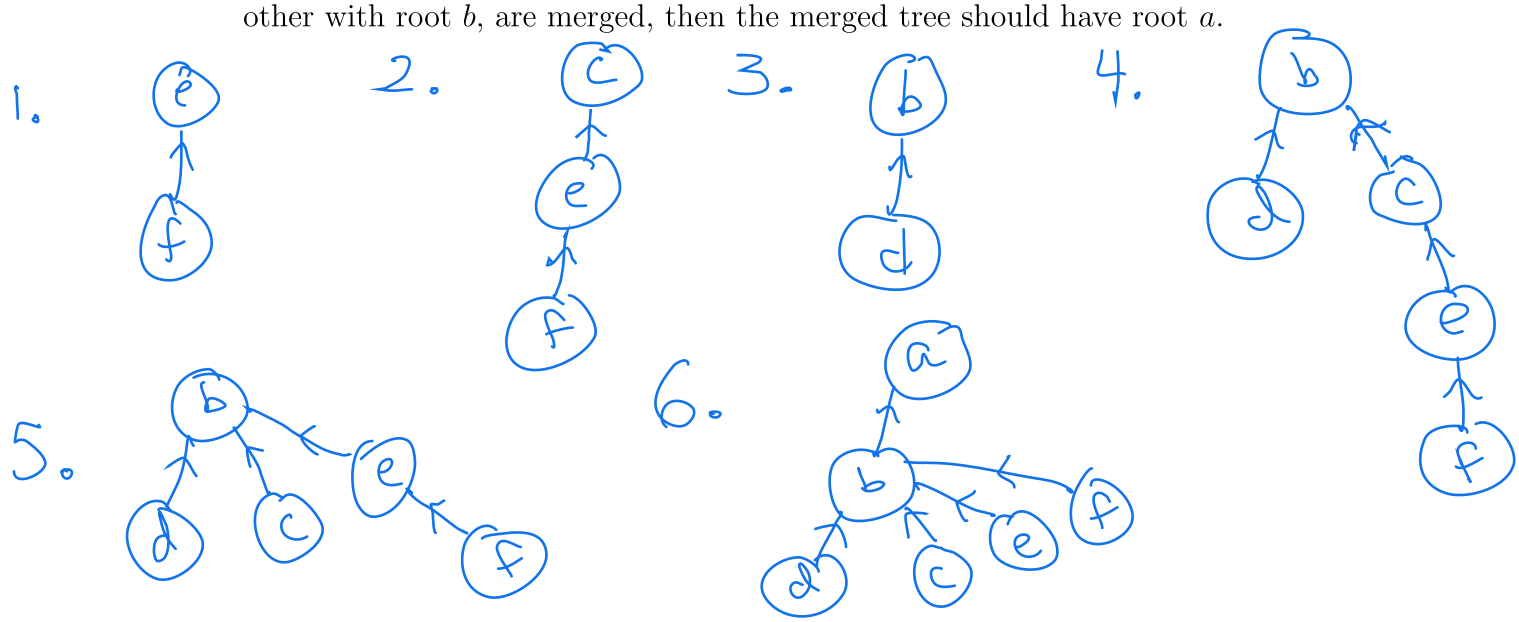
$u = cf + dh = \boxed{24} = -P_7 + P_5 - P_3 + P_1 = -10 + 42 - 3 - 5 = \boxed{24}$ ✓

LO3. Do the following.

(a) For the weighted graph with edges

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

Show how the membership-tree forest (not the Kruskal forest!) changes when processing each edge in the Kruskal sorted order when performing Kruskal's algorithm. When merging (i.e. unioning) two trees, use the convention that the root of the merged tree should be the one having *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are merged, then the merged tree should have root a.



(b) Demonstrate the greedy algorithm for `Unit-Task Scheduling (UTS)` for the `UTS` instance shown below.

Task	a	b	c	d	e	f	g	h	i	j	k	l
Deadline	2	2	4	5	1	8	5	1	3	8	2	4
Profit	60	10	30	60	20	50	20	30	50	50	40	20

Sorted Tasks: a, d, f, i, j, k, c, h, e, g, l, b

k	a	i	c	d		j	f
1	2	3	4	5	6	7	8

LO4. Do the following.

(a) The dynamic-programming algorithm that solves the `0-1 Knapsack` problem defines a function $p(i, c)$. Describe in one sentence the definition of $p(i, c)$ and give the ranges for i and c. Do *not* provide a recurrence (part b will ask for this).

See Lecture Notes

(b) Provide the dynamic-programming recurrence for $p(i, c)$.

See Lecture Notes

(c) Apply the recurrence from Part b to the items 1−6 whose respective weights are 3, 3, 4, 2, 5, 2 and respective profits are 20, 50, 30, 10, 20, 50. Assume a knapsack capacity equal to $M = 10$.

$P(i, c)$

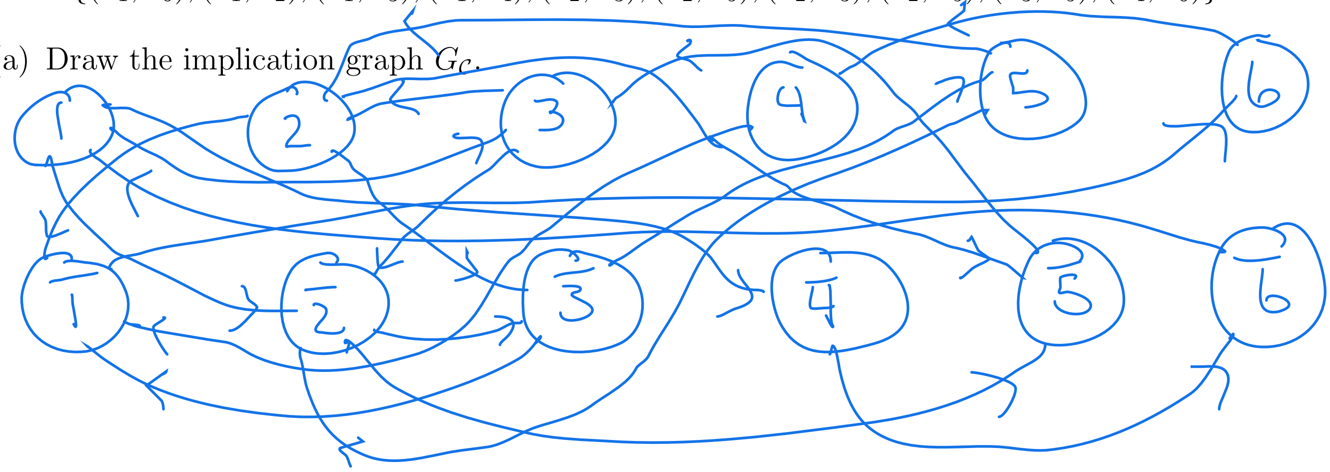
i \ c	0	1	2	3	4	5	6	7	8	9	10
0	0	$\boxed{0}$	0	0	0	0	0	0	0	0	0
1	0	$\boxed{0}$	0	20	20	20	20	20	20	20	20
2	0	0	0	50	$\boxed{50}$	50	70	70	70	70	70
3	0	0	0	50	50	50	70	80	$\boxed{80}$	80	100
4	0	0	10	50	50	60	70	80	$\boxed{80}$	90	100
5	0	0	10	50	50	60	70	80	$\boxed{80}$	90	100
6	0	0	50	50	60	100	100	110	120	130	$\boxed{130}$

Optimal Knapsack: {2, 3, 6}

LO5. Consider the 2SAT instance

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

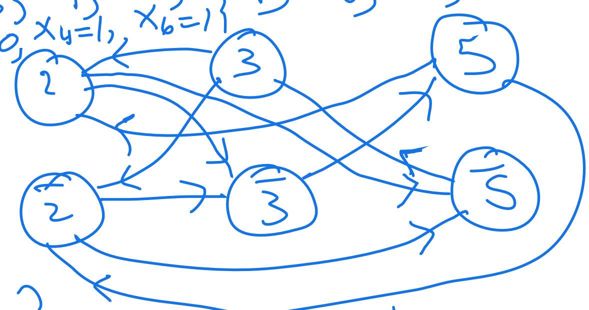
(a) Draw the implication graph $G_{\mathcal{C}}$.



(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.

$R_{x_1} = \{x_1, \overline{x}_2, x_3, \overline{x}_4, \overline{x}_3, \overline{x}_1, x_2, x_4, \overline{x}_6, x_6, x_5, \overline{x}_5\}$ is incons.

$R_{\overline{x}_1} = \{\overline{x}_1, x_6, x_4\}$ $\alpha = (x_1 = 0, x_4 = 1, x_6 = 1)$ $R_{\overline{x}_1}$



$R_{x_2} = \{x_2, \overline{x}_3, \overline{x}_5, x_3, \overline{x}_2, x_5\}$ is inconsistent

$R_{\overline{x}_2} = \{\overline{x}_2, \overline{x}_5, \overline{x}_3, x_5, x_3, x_2\}$ is inconsistent.

$\therefore$ $\mathcal{C}$ is unsatisfiable

(c) Suppose 2SAT instance $\mathcal{C}$ has three variables and, when running the original 2SAT algorithm on input $\mathcal{C}$, the algorithm terminates after exactly three queries to the reachability oracle and returns the value 0. If the first query/answer is $\texttt{reachable}(G_{\mathcal{C}}, x_1, \overline{x}_1) = 0$, provide the other two queries along with their answers. Explain your reasoning.

Reachable$(G_{\mathcal{C}}, x_2, \overline{x}_2) = 1$

Reachable$(G_{\mathcal{C}}, \overline{x}_2, x_2) = 1$

To prove unsatisfiability one needs two "Yes" answers for Reachable$(G_{\mathcal{C}}, x_i, \overline{x}_i)$ and Reachable$(G_{\mathcal{C}}, \overline{x}_i, x_i)$ for some i. We must have $i = 2$, since, e.g. Reachable$(G_{\mathcal{C}}, x_2, \overline{x}_2) = 0$ means only one more query for variable x_3.