CECS 528, Quiz 1 Solutions, Fall 2025, Dr. Ebert

IMPORTANT: READ THE FOLLOWING DIRECTIONS. Directions,

- For each problem, write your solution using **ONE SHEET OF PAPER ONLY (BOTH FRONT AND BACK)**. Write **NAME** and **PROBLEM NUMBER** on each sheet.
- Write solutions to different problems on **SEPARATE SHEETS** of paper.
- You may solve **at most TWO** Make up problems: either 0.5, 1, $0.5+0.5+0.5$, $1+0.5$, $1+1$, or $1+0.5+0.5$.

Unit 2 LO Problems

LO5. Do the following.

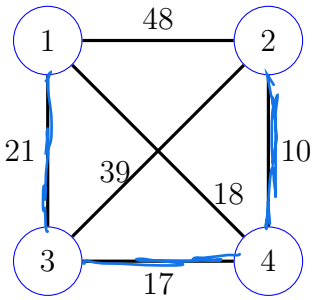
(a) The dynamic-programming algorithm that solves the `Runaway Traveling Salesperson` optimization problem defines a recurrence for the function $\text{mc}(i, A)$. In words, what does $\text{mc}(i, A)$ equal? Hint: do *not* write the recurrence (see Part b).

Solution. $\text{mc}(i, A)$ is the least cost required to start at vertex i and visit every vertex in A exactly once.

(b) Provide the dynamic-programming recurrence for $\text{mc}(i, A)$.

Solution. See exercise solution from Dynamic Programming lecture.

(c) Apply the recurrence from Part b to the graph below in order to calculate $\text{mc}(1, \{2, 3, 4\})$ Show all the necessary computations and use the solutions to compute an optimal path for the salesperson.



mc(1,{2,3,4}) =
min(48 + mc(2,{3,4}), 18 + mc(4,{2,3}),
21 + mc(3,{2,4})) =
min(48+27, 18+49, 21+27) =
i=2 i=4 i=3

Solution.

mc(2,{3,4}) = min(39+17, 10+17) = 27 i=4
i=3 i=4

mc(4,{2,3}) = min(10+39, 17+39) = 49 i=2
i=2 i=3

mc(3,{2,4}) = min(39+10, 17+10) = 27 i=4
i=2 i=4

48 i=3

opt. path:

P = 1, 3, 4, 2

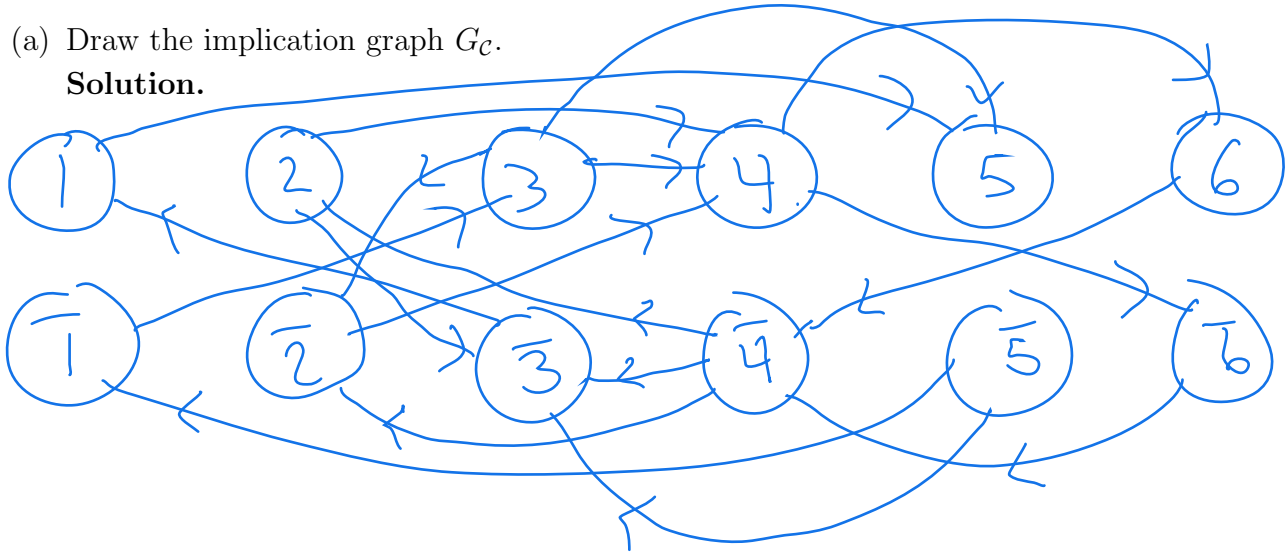
cost(P) = 48

LO6. Consider the 2SAT instance

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

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

Solution.

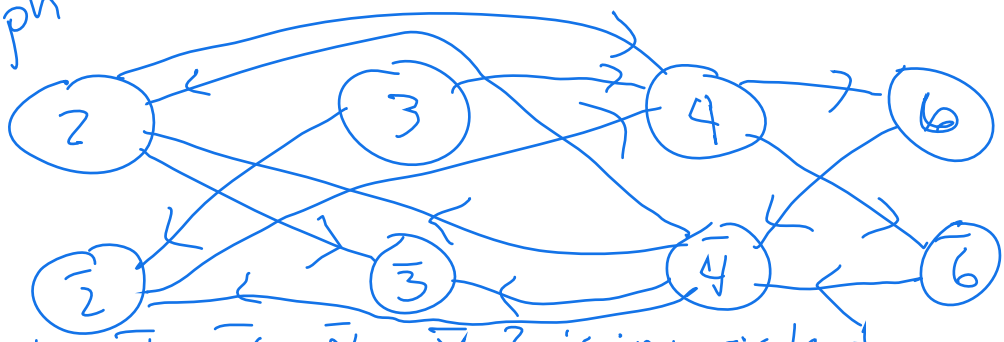


(b) Perform the Improved 2SAT algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced 2SAT instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable.

Solution.

$R_{x_1} = \{x_1, x_5\}$ is consistent. $\sigma_{R_{x_1}} = (x_1 = 1, x_5 = 1)$

Reduced graph



$R_{x_2} = \{x_2, x_4, x_6, \overline{x}_6, \overline{x}_4, \overline{x}_3, \overline{x}_2\}$ is inconsistent

$R_{\overline{x}_2} = \{\overline{x}_2, x_4, x_6, \overline{x}_6, \overline{x}_4, x_2, \overline{x}_3\}$ is inconsistent

$\therefore \mathcal{C}$ is unsatisfiable

(c) Suppose 2SAT instance $\mathcal{C}$ is satisfiable and has the unique satisfying assignment $\alpha = (x_1 = 1, x_2 = 0, x_3 = 1)$. If the original 2SAT algorithm is run with $\mathcal{C}$ as input, then at most how many queries will the algorithm make to the Reachability oracle? Explain. Hint. For each i, assume that the query reachable$(G_{\mathcal{C}}, x_i, \overline{x}_i)$ precedes the query reachable$(G_{\mathcal{C}}, \overline{x}_i, x_i)$.

Solution. According to the original 2SAT Algorithm Reach$(G_{\mathcal{C}}, x_1, \overline{x}_1) = 0$ and Reach$(G_{\mathcal{C}}, x_3, \overline{x}_3) = 0$ will nullify the need for Reach$(G_{\mathcal{C}}, \overline{x}_1, x_1)$ and Reach$(G_{\mathcal{C}}, \overline{x}_3, x_3)$

0% Either 3 or 4 queries total, Note: it's possible that Reach(Ge, X_3, $\bar{X}_3$) may either equal 0 or 1.

Makeup Problems

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 3T(n/4) + n^{\frac{3}{2}}$. Defend your answer.

Solution. $n^{\frac{3}{2}} = \Omega(n^{\log_4 3 + \varepsilon})$, for $\varepsilon = 0.5$. This is true since $\log_4 3 < 1$ and adding 0.5 to it yields a value less than $3/2 = 1.5$.

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

$$T(n) = 2T(n/4) + \sqrt{n},$$

then $T(n) = \Omega(\sqrt{n} \log n)$.

Solution. Inductive assumption: $T(k) \geq c\sqrt{k}\log k$ for all $k < n$ and some constant $c > 0$. Show $T(n) \geq \sqrt{n}\log n$.

$T(n) = 2T(n/4) + \sqrt{n} \geq 2c\sqrt{n/4}\log(n/4) + \sqrt{n} =$

$c\sqrt{n}(\log n - 2) + \sqrt{n} = c\sqrt{n}\log n - 2c\sqrt{n} + \sqrt{n}$

$\geq c\sqrt{n}\log n \iff 2c\sqrt{n} \leq \sqrt{n} \iff c \leq \frac{1}{2}$

LO2. Solve the following problems.

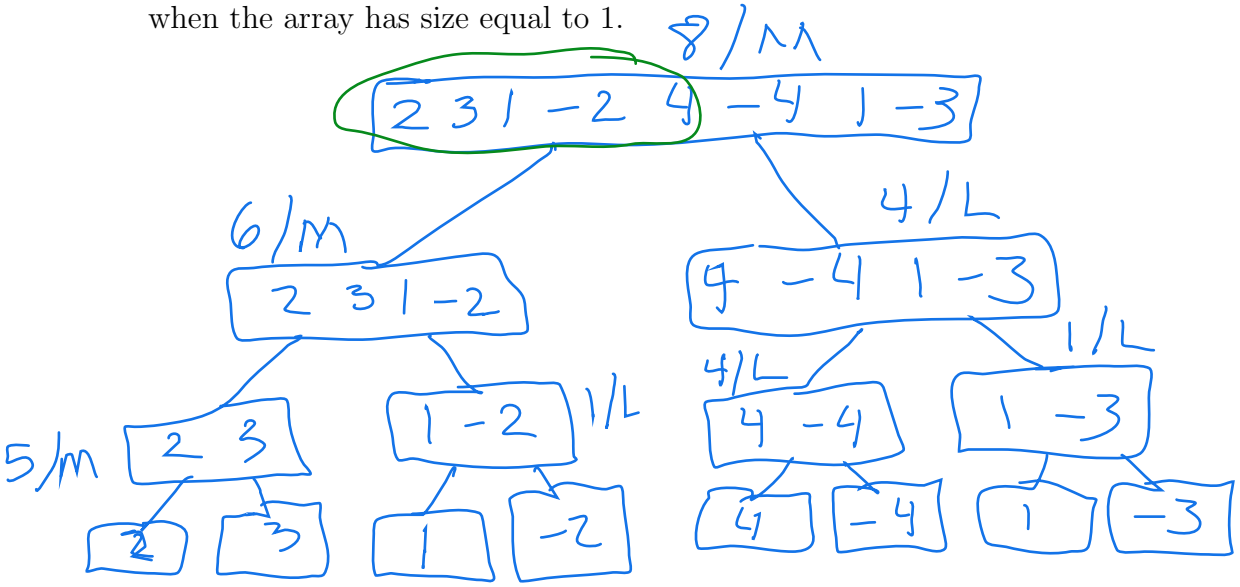
(a) Consider Karatsuba's algorithm which we'll call `multiply` for multiplying two even-length n-bit binary numbers x and y. Let x_L and x_R be the leftmost $n/2$ and rightmost $n/2$ bits of x respectively. Define y_L and y_R similarly. Let P_1 be the result of calling `multiply` on inputs x_L and y_L, P_2 be the result of calling `multiply` on inputs x_R and y_R, and P_3 the result of calling `multiply` on inputs $x_L + x_R$ and $y_L + y_R$. Then return the value

$$P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2.$$

Provide the divide-and-conquer recurrence that is satisfied by the function $T(n)$, where $T(n)$ denotes the worst-case number of steps required by the algorithm. Justify each component. Hint: there are two essential points that must be made in order to justify the choice for $f(n)$.

Solution. $T(n) = 3T(n/2)+n$ since three problems of size $n/2$ must be recursively solved. Moreover, dividing the problem instance and combining the three subproblem solutions requires adding two $O(n)$-bit numbers a finite number of times, and thus requires $O(n)$ total steps. Also, multiplying by 2^n and $2^{n/2}$ can be done by left-bit-shifting the binary number by $O(n)$-bits which can be done in $O(n)$ steps.

(b) For the `Maximum Subsequence Sum (MSS)` divide-and-conquer algorithm described in the Divide and Conquer core exercises, provide the entire recursion tree for instance $a = 2, 3, 1, -2, 4, -4, 1, -3$. Label each node with the problem instance that is being solved at that stage of the recursion. Also, next to each node write its MSS value and (in case the node is internal) indicate if the MSS comes from the left subproblem, right subproblem, or middle of both subproblems. For example, iff the MSS for some subproblem equals 7 and that value came from the left subproblem, then write 7/L. Assume a base case occurs when the array has size equal to 1.



LO3. Do the following.

(a) Consider the FFT algorithm when applied to a polynomial $A(x)$ having degree $2^n - 1$. Provide the equation that relates $A(x)$ to the two subproblem polynomials $A_e(x)$ and $A_o(x)$. What are the degrees of these two polynomials? Based on this equation, why is it essential that, for even n, the nth roots of unity come in additive-inverse pairs?

Solution. $A(x)$ has degree $2^n - 1$. $A(x) = A_e(x^2) + xA_o(x^2)$, where A_e and A_o are $(2^{n/2} - 1)$-degree polynomials.

(b) If $p(x) = -2 + 3x + x^2 - 5x^3$, then compute $\mathrm{DFT}^{-1}(p)$ using the FFT algorithm. Show the entire recursion tree as was done in the lecture notes. (10 points)

$DFT_4^{-1}(-2,3,1,-5) =$

$\frac{1}{2}\left[\left(-\frac{1}{2},-\frac{3}{2},-\frac{1}{2},-\frac{3}{2}\right) + (1,-i,-1,i) \odot (-1,4,-1,4)\right] = \left(\frac{-3}{4}, \frac{-3}{4} - 2i, \frac{1}{4}, -3+2i\right)$

$DFT_2^{-1}(-2,1) =$

$\frac{1}{2}\left[(-2,-2) + (1,-1) \odot (1,1)\right] =$

$\left(-\frac{1}{2}, -\frac{3}{2}\right)$

$DFT_2^{-1}(3,-5) =$

$\frac{1}{2}\left[(3,3) + (1,-1) \odot (-5,-5)\right] =$

$(-1,4)$

$DFT_1^{-1}(-2) = -2$ $DFT_1^{-1}(1) = 1$

$DFT_1^{-1}(3) = 3$ $DFT_1^{-1}(-5) = -5$

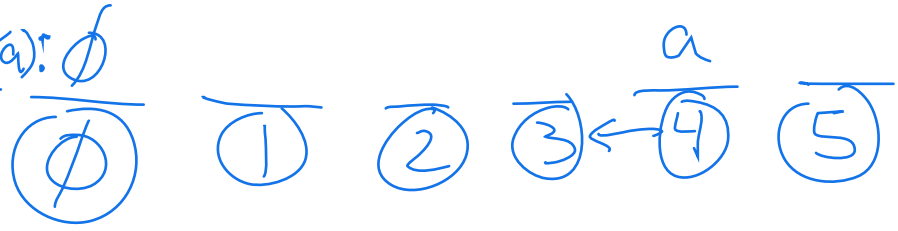
LO4. Do the following.

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

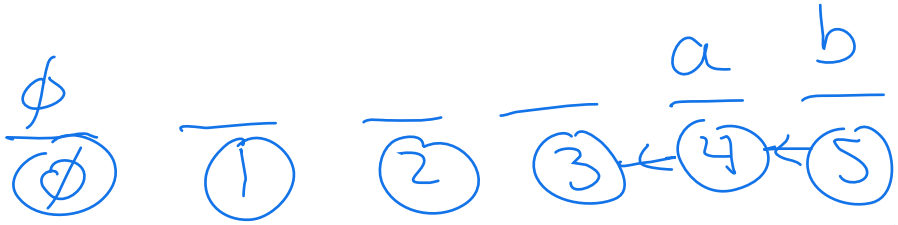
Task	a	b	c	d	e	f
Deadline Index	4	5	4	5	3	4
Profit	60	50	40	30	20	10

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). Note that the earliest possible deadline index is 1, meaning that the earliest slot in the schedule array has index 1. Also, assume that an insert attempt that takes place at index i results in the function call `root`(i), followed by a `union` operation. Finally, to receive credit, your solution should show six different snapshots of the M-Tree forest.

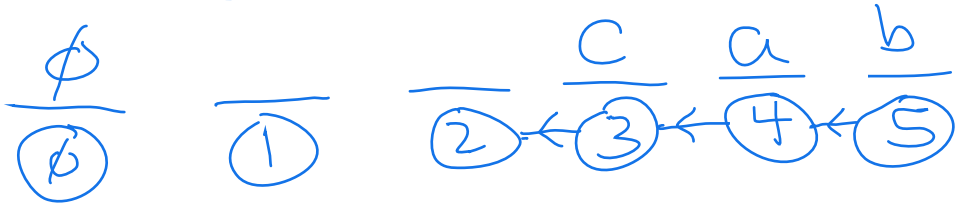
Ins(a): ∅



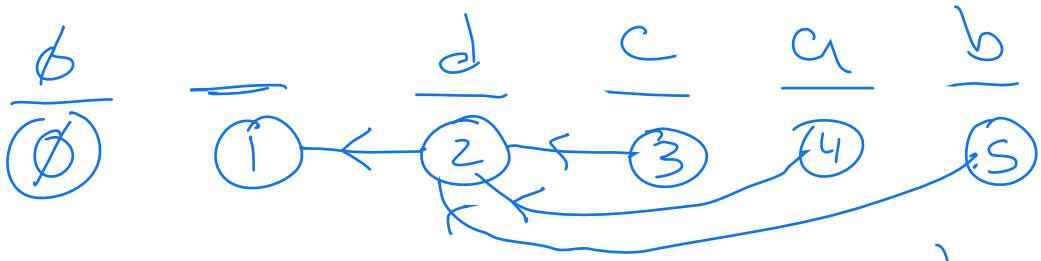
Insert b:



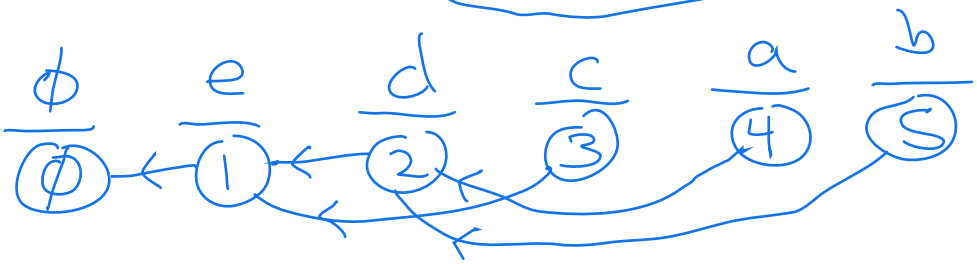
Insert c:



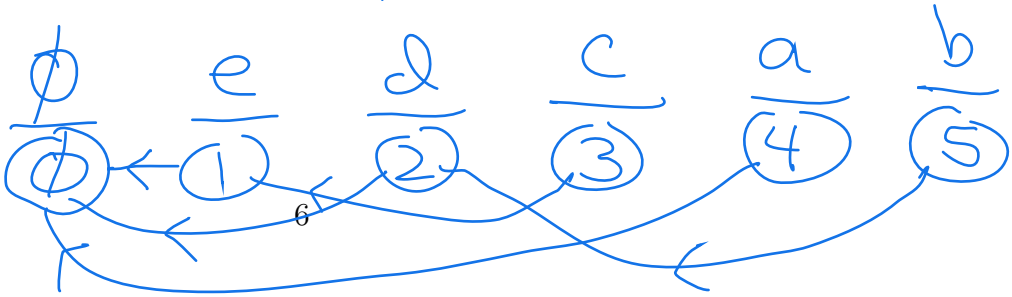
Insert d:



Insert e:



Insert f (fail):



(b) An instance of the `Unit Task Scheduling` problem consists of the tasks shown in the table below.

Task	a	b	c	d	e	f	g	h	i	j	k
Deadline	4	8	6	6	4	2	3	2	8	7	1
Profit	50	80	30	30	70	30	70	50	80	20	60

i. What is the greedy choice that is being made during each step of the algorithm?
Solution. The next task to be schedule is the one that yields the most profit and for which there is an open slot i for which $i \leq d$, where d is the task's deadline.

ii. Apply the greedy choice successively to obtain an optimal schedule. Show work.