CECS 528, Learning Outcome Assessment 8, Spring 2024, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each solution on a separate sheet of paper.

Problems

LO4. Answer/solve the following.

(a) The FFT algorithm owes its existence to what two properties that are possessed by the nth roots of unity when n is even?

Solution. The roots come in $n/2$ different additive-inverse pairs and the squares of the roots yield the $n/2$-roots of unity.

(b) Compute $\mathrm{DFT}_4^{-1}(5, -4, 3, -2)$ using the IFFT algorithm. Show the solution to each of the seven subproblem instances and, for each one, clearly represent it using DFT notation and apply the formula for computing it. Show all work.

Solution.

$\mathrm{DFT}_4^{-1}(5,-4,3,-2) = \frac{1}{2}[(4,1,4,1) + (1,-i,-1,i) \odot (-3,-1,-3,-1)] = (\frac{1}{2}, \frac{1+i}{2}, \frac{7}{2}, \frac{1-i}{2})$

$\mathrm{DFT}_2^{-1}(5,3) = \frac{1}{2}((5,5) + (1,-1) \odot (3,3)) = (4,1)$

$\mathrm{DFT}_2^{-1}(-4,-2) = \frac{1}{2}[(-4,-4) + (1,-1) \odot (-2,-2)] = (-3,-1)$

$\mathrm{DFT}_1^{-1}(5) = 5$ $\mathrm{DFT}_1^{-1}(3) = 3$

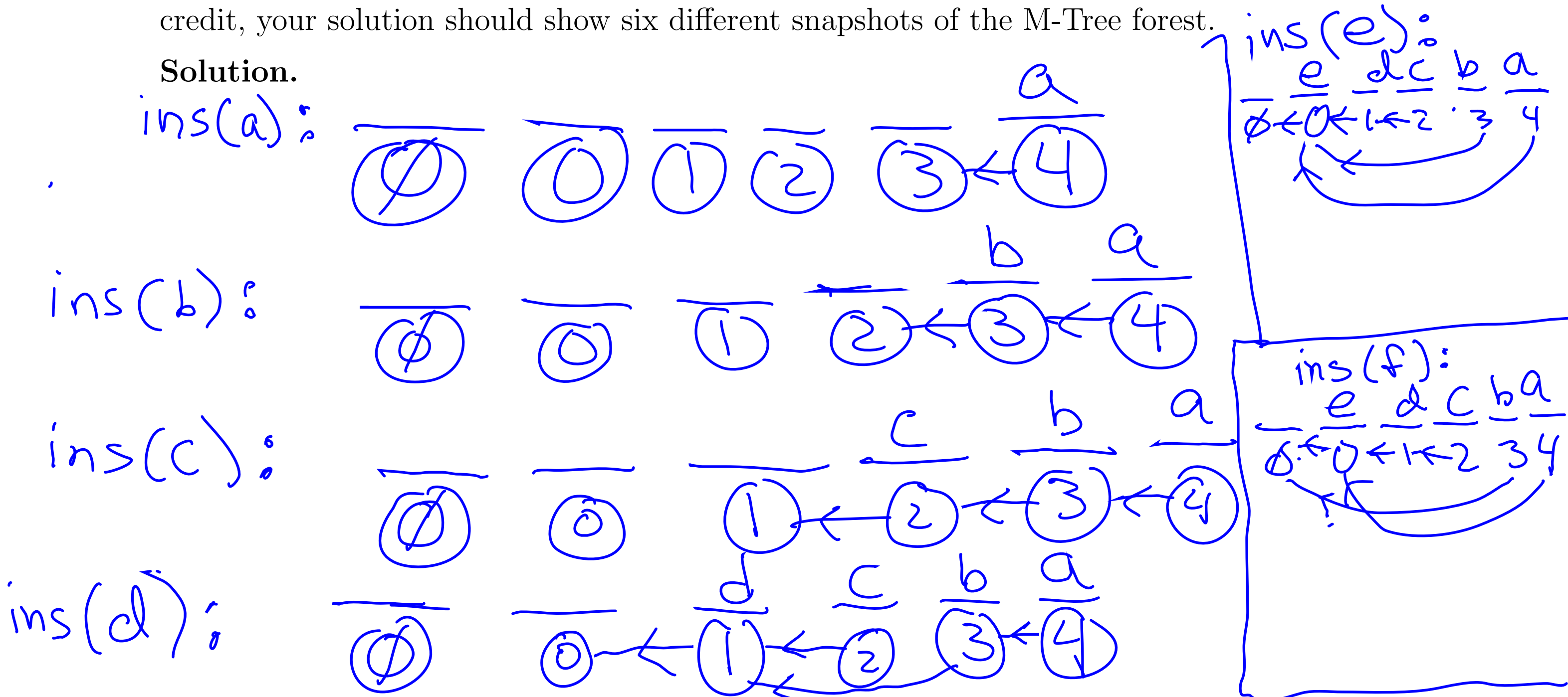
$\mathrm{DFT}_1^{-1}(-4) = -4$ $\mathrm{DFT}_1^{-1}(-2) = -2$

LO5. 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	4	3	3	4	3
Profit	60	50	40	30	20	10

For each task, 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 0, meaning that the earliest slot in the schedule array has index 0. Also, assume that an insert attempt that takes place at index i results in the function call `root`(i). Finally, to receive credit, your solution should show six different snapshots of the M-Tree forest.

Solution.



LO6. Answer the following with regards to a correctness-proof outline for the Fractional Knapsack algorithm.

(a) Assume $x_1, x_2, \ldots, x_n$ is an ordering of the items in *decreasing* order of profit density (i.e. profit per unit weight). Let $f_i \in [0, 1]$ denote the fraction of item x_i that the FK-algorithm adds to the knapsack, $i = 1, 2, \ldots, n$. Explain why $f_1 \geq f_2 \geq \cdots \geq f_n$ is a non-increasing sequence of fractions.

Solution. Since the greedy algorithm attempts to add all of item x_i, we see that there is a k for which

$$f_1 = f_2 = \cdots = f_{k-1} = 1,$$

$f_k \leq 1$, and $f_l = 0$ for all $l > k$. Thus, $f_1 \geq f_2 \geq \cdots \geq f_n$.

(b) Let $f'_1, f'_2, \ldots, f'_n$ be a sequence of fractions that optimizes total profit, and assume that $f_i = f'_i$, for all $i < k$, but $f_k \neq f'_k$. Explain why, in this case, it must be true that $f'_k < f_k$. Hint: what is the contradiction in case the opposite was true?

Solution. Since the greedy algorithm attempts to add as much of x_k as possible, it must be the case that $f_k > f'_k$.

(c) From part b, the optimal solution uses $(f_k - f'_k)w_k$ less weight of item x_k. Suppose it uses $(f_k - f'_k)w_k$ *more* weight of item x_{k+1} than does FKA. Show that the FKA solution will earn at least as much profit on items $x_1, \ldots, x_k, x_{k+1}$ as the optimal solution will earn on these same items. In other words, show that the difference between the FKA total profit and the optimal total profit is nonnegative. Why does this imply that both total profits are equal?

Solution. Since the profit density d_k satisfies $d_k \geq d_{k+1}$, then the profit earned by the greedy algorithm for items x_k and x_{k+1} satisfies

$$d_k f_k w_k + d_{k+1} f_{k+1} w_{k+1} \geq d_k f'_k w_k + d_{k+1}(f_{k+1} + f_k - f'_k) w_{k+1}.$$

Verify!

LO7. Answer the following.

(a) Provide the dynamic-programming recurrence for computing the maximum-cost path, denoted $\mathrm{mc}(u, v)$, from a vertex u to a vertex v in a directed acyclic graph (DAG) $G = (V, E, c)$, where $c(x, y)$ gives the cost of edge $e = (x, y)$, for each $e \in E$. The recurrence should allow one to compute the maximum costs from a single source to all other vertices in a linear number of steps. Hint: step backward from v.

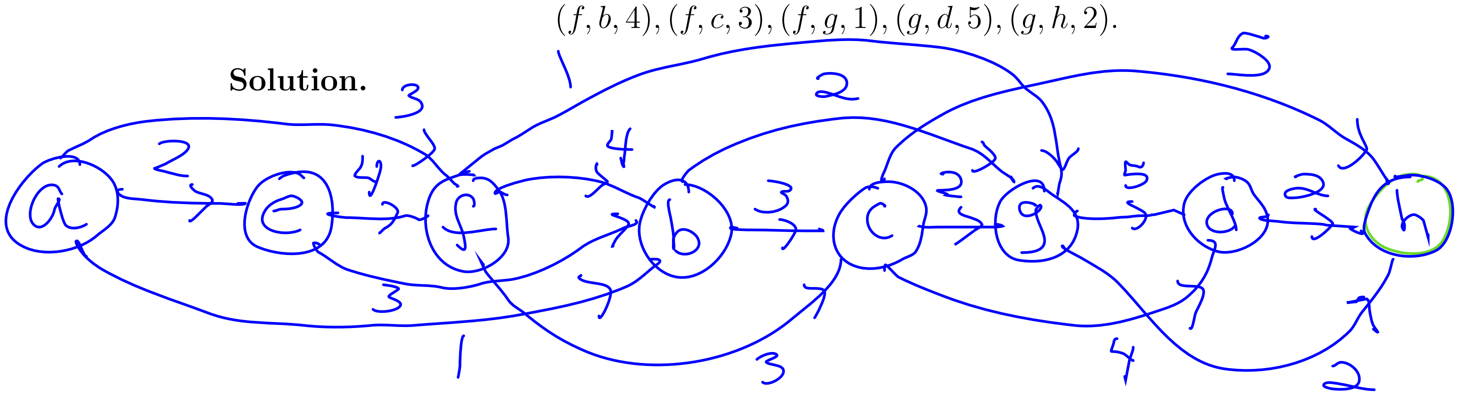
Solution.

$$\mathrm{mc}(u,v) = \begin{cases} 0 & \text{if } u = v \\ \infty & \text{if } \deg^+(v) = \infty \\ \max\limits_{(w,v)\in E} (\mathrm{mc}(u,w) + c(w,v)) & \end{cases}$$

(b) Draw the vertices of the following DAG G in a linear left-to-right manner so that the vertices are topologically sorted, meaning, if (u, v) is an edge of G, then u appears to the left of v. The vertices of G are a-h, while the weighted edges of G are

$$(a, b, 1), (a, e, 2), (a, f, 3), (b, c, 3), (b, g, 2), (c, d, 4), (c, g, 2), (c, h, 5), (d, h, 2), (e, b, 3), (e, f, 4),$$
$$(f, b, 4), (f, c, 3), (f, g, 1), (g, d, 5), (g, h, 2).$$

Solution.



$\mathrm{mc}(a,a) = 0$ $\mathrm{mc}(a,e) = 2$ $\mathrm{mc}(a,f) = \max(\mathrm{mc}(a,a)+3, \mathrm{mc}(a,e)+4) = 6$ $\mathrm{mc}(a,b) = \max(1+\mathrm{mc}(a,a), \mathrm{mc}(a,e)+3, \mathrm{mc}(a,f)+4)$
$= \max(1, 5, 6+4) = 10$

$\mathrm{mc}(a,c) = \max(\mathrm{mc}(a,b)+3, \mathrm{mc}(a,f)+3) = 13$

$\mathrm{mc}(a,g) = \max(\mathrm{mc}(a,b)+2, \mathrm{mc}(a,f)+1, \mathrm{mc}(a,c)+2) = 15$

$\mathrm{mc}(a,d) = \max(\mathrm{mc}(a,g)+5, \mathrm{mc}(a,c)+4) = 20$

(c) Starting from left to right in topological order, use the recurrence to compute

$$\mathrm{mc}(a,a), \ldots, \mathrm{mc}(a,h).$$

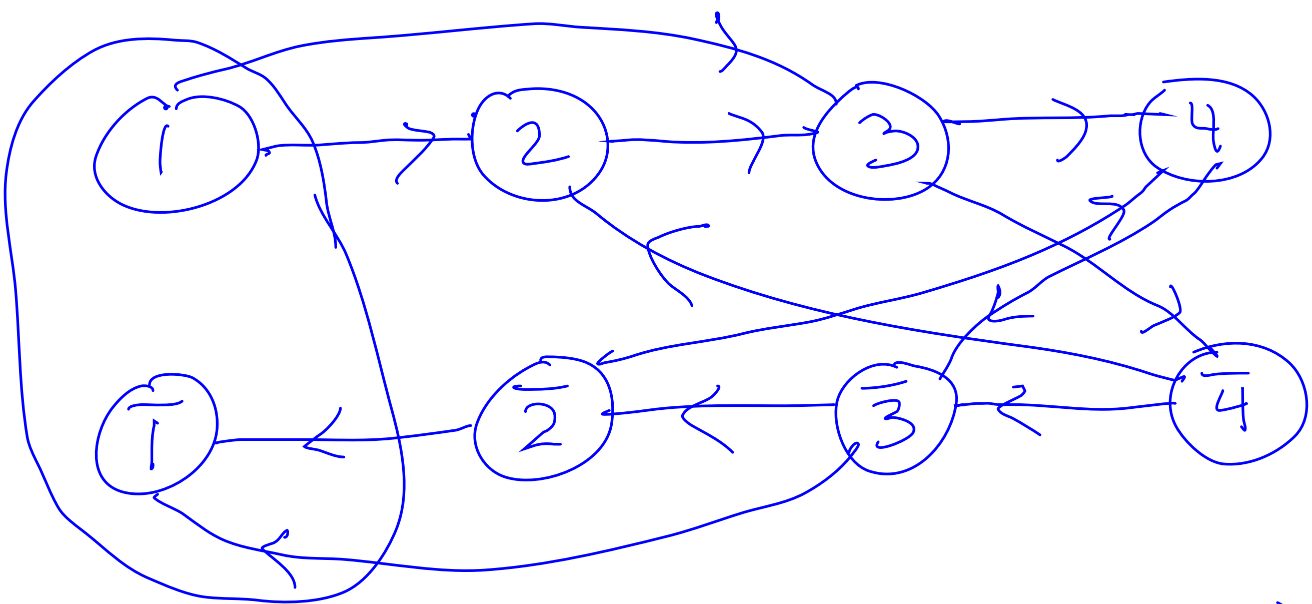
Solution. $\mathrm{mc}(a,h) = \max(\mathrm{mc}(a,c)+5, \mathrm{mc}(a,g)+2, \mathrm{mc}(a,d)+2) = \max(18, 17, 22) = 22.$

LO8. Do/answer the following.

(a) Draw the implication graph $G_{\mathcal{C}}$ associated with the 2SAT instance

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

Solution.



$R_1 = \{1, 3, \overline{4}, 2, 4, \overline{3}, \overline{2}, \overline{1}\}$

$R_{\overline{1}} = \{\overline{1}\}$ $R_2 = \{2, \overline{2}, 3, \overline{3}, 4, \overline{4}\}$

$R_{\overline{2}} = \{\overline{2}, 4, \overline{3}\}$

(b) Apply the improved `2SAT` algorithm to obtain a satisfying assignment for $\mathcal{C}$. When deciding on the next reachability set R_l to compute, follow the literal order $l = x_1, \overline{x}_1, \ldots, x_4, \overline{x}_4$. For each consistent reachability set encountered, provide the partial assignment α_{R_l} associated with R_l and draw the reduced implication graph before continuing to the next reachability set. Note: do *not* compute the reachability set for a literal that has already been assigned a truth value. Provide a final assignment α and verify that it satisfies all six clauses.

Solution.

$\alpha = \alpha_{R_{\bar{1}}} \cup \alpha_{R_{\bar{2}}} = (x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 1)$

(c) Suppose a `Reachability`-oracle answers "yes" to the query reachable$(G_{\mathcal{C}}, \overline{x}_2, x_2)$. If $\mathcal{C}$ is satisfiable via assignment α, then what is the value of $\alpha(x_2)$? Explain.

Solution. $\alpha(x_2) = 1$ since R_{x_2} is a consistent reachability set. Why? Since $\mathcal{C}$ is satisfiable and $R_{\overline{x}_2} \supseteq \{x_2, \overline{x}_2\}$ is inconsistent.