CECS 528, Learning Outcome Assessment Makeup Problems, Pink, December 6th 2023, Dr. Ebert

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

Problems

LO1. Complete the following problems.

(a) Show each of the subproblem instances that must be solved when using the recursive division algorithm for finding the quotient and remainder of x/y. Do this for $x = 152$ and $y = 19$. Make sure to provide the solution to each subproblem instance. Hint: there are nine subproblem instances, including the original problem instance.

(b) For the Strassen-Solovay primality test with $n = 21$, determine whether or not $a = 2$ is a witness to n not being prime. Do this by evaluating *both* sides of the test congruence, mod 21.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 4T(n/2) + n^{\log_4 8}$. Defend your answer.

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

$$T(n) = 4T(n/2) + 4n^{2.5}$$

then $T(n) = \mathrm{O}(n^{2.5})$.

LO2. Solve the following problems.

(a) Recall the combine step of the `Minimum Distance Pair (MDP)` algorithm where, for each point P in the δ-strip, there is a $2\delta \times \delta$ rectangle whose bottom side contains P and is bisected by the vertical line that divides the points into left and right subsets. Explain why there can be at most 7 other points (from the problem instance) in this rectangle.

(b) Use Strassen's products $P_1 = a(f - h) = af - ah$, $P_2 = (a + b)h = ah + bh$, $P_3 = (c + d)e = ce + de$, $P_4 = d(g - e) = dg - de$, $P_5 = (a + d)(e + h) = ae + ah + de + dh$, $P_6 = (b - d)(g + h) = bg + bh - dg - dh$, and $P_7 = (a - c)(e + f) = ae - ce - cf + af$ to compute the matrix product

$$\begin{pmatrix} 4 & -1 \\ -3 & 6 \end{pmatrix} \begin{pmatrix} -3 & 4 \\ 2 & -5 \end{pmatrix}$$

Show all work. (13 pts)

LO3. Answer 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?

(b) Compute $\text{DFT}_4(5, -4, 3, -2)$ using the FFT 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.

LO4. 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.

(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?

(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?

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	2	3	0	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.

LO6. Answer the following.

(a) Provide the dynamic-programming recurrence for computing the distance $\text{D}(u, v)$, 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 $e \in E$. Hint: step *backward* from v, rather than forward from u.

(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

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

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

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

$$d(a, a), \ldots, d(a, h).$$

LO7. Do/answer the following.

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

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

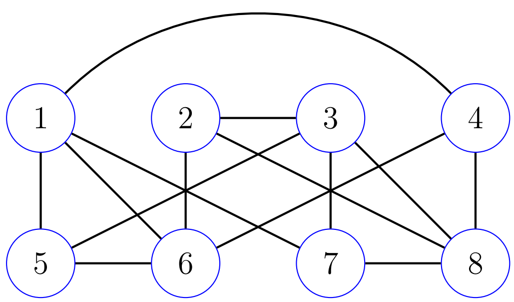
(b) Apply the improved `2SAT` algorithm to obtain a satisfying assignment for $\mathcal{C}$. When deciding on the next reachability set R_l to compute, follow the literal order $l = x_1, \overline{x}_1, \ldots, x_5, \overline{x}_5$. For each consistent reachability set encountered, provide the partial 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 the clauses.

(c) In the orginal `2SAT` algorithm, what is the least number of queries to `Reachability` oracle that is needed before one can conclude that an instance of `2SAT` is unsatisfiable? Explain.

LO8. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction fromm problem A to problem B. Hint: do *not* assume that A and B are decision problems.

(b) The simple graph $G = (V, E)$ shown below is an instance of the `Maximum Independent Set (MIS)` optimizationn problem. Draw $f(G)$, where f is the mapping reduction from `MIS` to `Maximum Clique` provided in lecture.



(c) Verify that f is valid for input G in the sense that both G and $f(G)$ have the same solution. Hint: for both problems please assume that a "solution" is represented by a subset of vertices.

LO9. Recall that an instance of the `Vertex Cover` decision problem is a pair (G, k), where $G = (V, E)$ is a simple graph, $k \geq 0$ is an integer, and the problem is to decide if G has a **vertex cover** of size k, i.e. a set $C \subseteq V$ for which every edge $e \in E$ is incident with at least one vertex in C.

(a) For a given instance (G, k) of `Vertex Cover` describe a certificate in relation to (G, k).

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance (G, k), ii) a certificate for (G, k) as defined in part a, and decides if the certificate is valid for G.

(c) Provide size parameters that may be used to measure the size of an instance of `Vertex Cover`.

(d) Use the size parameters from part c to describe the running time of your verifier from part b. Defend your answer in relation to the algorithm you provided for the verifier.

LO10. Recall the mapping reduction from `SAT` to `3SAT` described in lecture.

(a) Given the `SAT` instance $F(x_1, x_2, x_3) = x_1 \wedge (\overline{x}_2 \vee x_3)$, draw its parse tree and provide the associated Boolean formula G that is satisfiability equivalent to F and serves as the beginning step of the reduction. Hint: formula G introduces y-variables.

(b) Rewrite formula G by making use of the logical identity

$$(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$$

(c) Rewrite the formula from part b by making use of the logical identity

$$(P \rightarrow Q) \Leftrightarrow (\overline{P} \vee Q).$$

(d) Rewrite the formula from part c by performing one or more applications of De Morgan's rule.

(e) Rewrite the formula from part d by performing one or more applications of the distributive rule in order to obtain an AND of OR's. Then convert the AND of OR's to an AND of ternary (i.e. three) OR's and use `3SAT` notation to complete the reduction.