CECS 528, Learning Outcome Assessment 12b, May 5th, Spring 2023, Dr. Ebert

Problems

LO1. Suppose function $f(n)$ defined as follows.

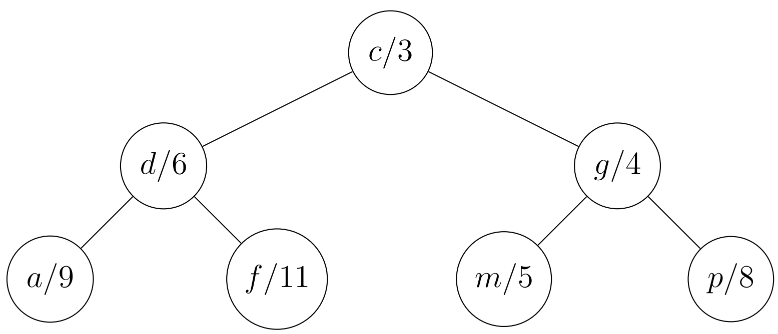
$$f(n) = \begin{cases} 4n^{2.3}\log^3 n & \text{if } n \bmod 3 = 0 \\ 2^{\log^{2.2} n} & \text{if } n \bmod 3 = 1 \\ 10n^{2.2}\log^{35} n & \text{if } n \bmod 3 = 2 \end{cases}$$

Provide a big-O upper bound and big-Ω lower bound for $f(n)$.

LO2. The tree below shows the state of the binary min-heap at the beginning of some round of Dijkstra's algorithm, applied to some weighted graph G. If G has directed edges

$$(c, e, 7), (c, f, 3), (c, g, 6), (c, p, 2), (f, c, 2),$$

then draw a plausible state of the heap at the end of the round.



LO3. Answer the following with regards to a correctness-proof outline for the Task Selection algorithm. (0 pts)

(a) Let $S = t_1, \ldots, t_m$ be the set of non-overlapping tasks selected by TSA and sorted by finish time, i.e. $f(t_i) < f(t_{i+1})$ for all $i = 1, \ldots, m-1$. Let S_{opt} be an optimal set of tasks and assume that, for some $k \geq 1$, $t_1, \ldots, t_k \in S_{\text{opt}}$, but $t_{k+1} \notin S_{\text{opt}}$. Let $t' \in S_{\text{opt}}$ be the earliest task that follows t_k. What can we say about the relationship between $f(t_{k+1})$ and $f(t')$? Justify your answer.

(b) Suppose we have established the existence of an optimal set of tasks S_{opt} such that $S \subseteq S_{\text{opt}}$. To finish the proof, we must show that there cannot be a task in S_{opt} that is *not* in S. As a step in this direction, explain why there can be no task $t \in S_{\text{opt}}$ that lies between t_i and t_{i+1}, where i is any value from $\{1, \ldots, n-1\}$.

LO4. Given the recurrence $T(n) = 2T(n/2) + n\log n$, use the substitution method to prove $T(n) = \Omega(n\log^2 n)$. Remember to state the inductive assumption.

LO5. At the top level of recursion for the Median-of-Five Find Statistic algorithm, with

$$a = 84, 38, 54, 60, 10, 74, 22, 75, 57, 19, 27, 64, 35, 3, 80, 68, 72, 13, 87, 36, 6, 5$$

and $k = 10$ as inputs,

(a) What number will be chosen as the pivot for the partitioning step? Show work.

(b) At the next level (1) of recursion, which array will be examined: a_{left}? a_{right}? neither? Explain.

LO6. Given that $r = ae + bg$, $s = af + bh$, $t = ce + dg$, and $u = cf + dh$ are the four entries of AB, and Strassen's products are obtained from matrices

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

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

Compute $P_1, \ldots, P_7$ and use them to compute r, s, t, and u.

LO7. Answer/solve the following questions/problems.

(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 $u =$ acbcaa and $v =$ aabbca. Show the matrix of subproblem solutions and use it to provide an optimal solution.

LO8. Answer the following.

(a) Provide the dynamic-programming recurrence for computing $\mathrm{d}(u, v)$ the distance from vertex u to vertex v in a directed acyclic graph (DAG) $G = (V, E, c)$, where $c(e)$ gives the cost of edge e, for each $e \in E$.

(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

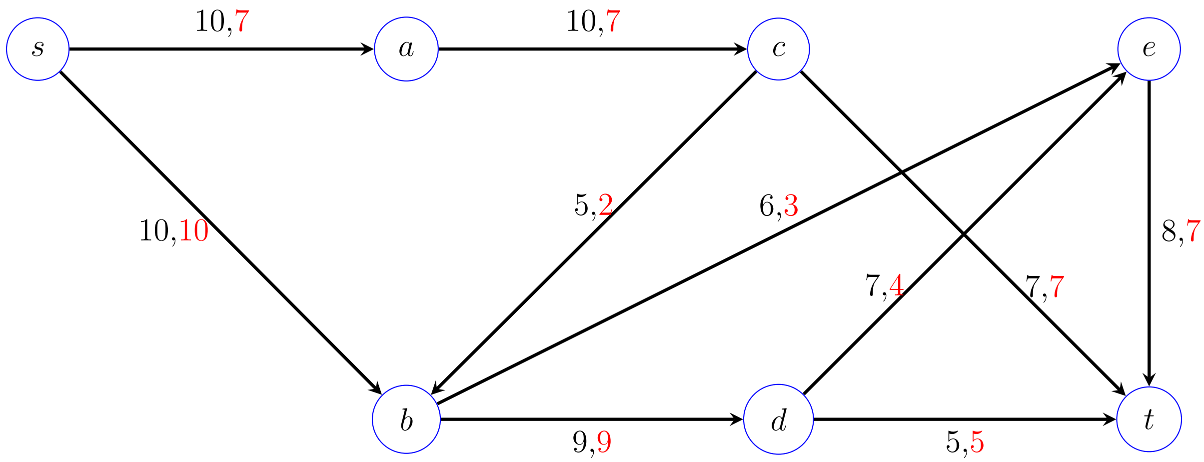
$$(a, b, 1), (a, e, 13), (a, f, 12), (b, c, 16), (b, g, 15), (c, d, 2), (c, g, 16), (c, h, 18), (d, h, 15), (e, b, 6)$$

$$(e, f, 19), (f, b, 19), (f, c, 7), (f, g, 4), (g, d, 15), (g, h, 12).$$

(c) Starting with $u = h$, and working backwards (from right to left in the topological sort), use the recurrence from part a to compute $d(u, h)$ for each $u \in \{a, b, \ldots, h\}$, where the ultimate goal is to compute $d(a, h)$.

LO9. A flow f (2nd value on each edge) has been placed in the network G below.

(a) Draw the residual network G_f and use it to determine an augmenting path P. Highlight path P in the network so that it is clearly visible.



(b) In the original network, cross out any flow value that changed, and replace it with its updated value from $f_2 = \Delta(f, P)$.

(c) What one query can be made to a `Reachability` oracle to determine if f_2 is a maximum flow for G? Hint: three inputs are needed for the `reachable` query function. Clearly define each of them.

LO10. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B.

(b) Is $f(n) = n^2 + 3n + 5$ a valid mapping reduction from the `Even` decision problem to the `Odd` decision problem? Justify your answer.

LO11. Recall the mapping reduction $f(F) = \mathcal{C}$, where f maps an instance of `SAT` to an instance of the `3SAT` decision problem. Given `SAT` instance

$$F(x_1, x_2, x_3) = \overline{x}_1 \vee (x_2 \wedge \overline{x}_3),$$

show each of the following steps towards computing $f(F)$.

(a) Draw F's parse tree, label its internal nodes with y-variables, and provide the initial Boolean formula that asserts that F is satisfiable.

(b) Convert the formula from part a to an equivalent one that uses only AND, OR, and NEGATION.

(c) Use De Morgan's rule and the distributive rule to convert your formula from part b to one that is an "AND of OR's".

(d) Convert the formula from part c to a `3SAT` instance by using `3SAT` notation, and duplicating literals whenever necessary in order to ensure that each clause has three literals.