CECS 528, Learning Outcome Assessment 5 (Version b), March 3rd, Spring 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to at most 2 LO problems on separate sheets of paper.

Problems

LO1. Determine the growth of the function

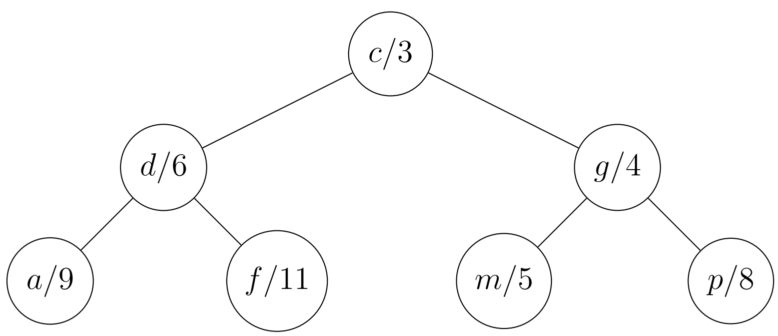
$$S(n) = \log 1 + 2\log 2 + \cdots + n \log n.$$

Show all work.

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.

(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. 6

(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 is any value from $\{1, \ldots, n-1\}$.

[t_i] [t] [t_{i+1}]

↑ $t \in S_{opt}$

LO4. Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 4T(n/2) + n^2,$$

Then $T(n) = \mathrm{O}(n^2 \log n)$. Hint: remember to state the inductive assumption.

LO5. Recall that the `Maximum Subsequence Sum (MSS)` problem (Exercise 34) admits a divide-and-conquer algorithm that, on input integer array a, requires computing the maximum of $\mathrm{MSS}_{\mathrm{left}}(a)$, $\mathrm{MSS}_{\mathrm{right}}(a)$, and $\mathrm{MSS}_{\mathrm{mid}}(a)$. If the array is

$$a = 4, -2, 5, -8, 4, 6, -5, 3, -8, 7$$

then

7 7

(a) Provide $\mathrm{MSS}_{\mathrm{left}}(a)$, $\mathrm{MSS}_{\mathrm{right}}(a)$. Hint: no need to show work!

(b) Provide the two arrays of sums, $\mathrm{Sum}_{\mathrm{left}}$ and $\mathrm{Sum}_{\mathrm{right}}$ that are used to compute $\mathrm{MSS}_{\mathrm{mid}}(a)$. Explain how to compute $\mathrm{MSS}_{\mathrm{mid}}(a)$ from these two arrays.

Left Sums = 4 −4 1 −1 3

Right Sums = 6 1 4 −4 3

$\mathrm{MSS}_{\mathrm{mid}} = 4 + 6 = 10$