CECS 528, Learning Outcome Assessment 3, 9/18/2024, Dr. Ebert

Directions: show all work.

Problem

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 7T(n/2) + n^2$. Defend your answer.

$n^2 = O(n^{\log 7 - \varepsilon})$ for $\varepsilon \le \log 7 - 2$

$\therefore$ By Case 1 of MT, $T(n) = \Theta(n^{\log 7})$

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

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

then $T(n) = \Omega(n^2)$. Assume $T(k) \ge ck^2$ for some $c > 0$ and all $k < n$.

Show $T(n) \ge cn^2$

$T(n) = 2T(n/2) + n^2 \ge 2c\left(\frac{n}{2}\right)^2 + n^2 = \frac{cn^2}{2} + n^2 \ge cn^2 \iff$

$\frac{cn^2}{2} \le n^2 \iff c \le 2$ ✓

LO2. Solve each of the following problems.

(a) Recall that the `Find-Statistic` algorithm makes use of the Partitioning algorithm and uses a pivot that is guaranteed to have at least

$$3(\lfloor \frac{1}{2}\lceil \frac{n}{5}\rceil \rfloor - 2) \ge 3(\frac{1}{2}\cdot\frac{n}{5} - 3) = \frac{3n}{10} - 9.$$

members of array a both to its left and to its right. Rewrite each of the above inequalities but now assuming the algorithm uses groups of nine instead of groups of five. Explain your reasoning for each of the numerical changes that you make.

$5 \Rightarrow 9$: Groups of 9 instead of 5

$3 \Rightarrow 5$: 5 members per group $\le$ (resp. $\ge$) pivot

$\therefore 5(\lfloor \frac{1}{2}\lceil \frac{n}{9}\rceil \rfloor - 2) \ge 5(\frac{n}{18} - 3) = \frac{5n}{18} - 15 \le \frac{n}{4}$ for n suff. large

(b) Recall that the `Maximum Subsequence Sum (MSS)` problem admits a divide-and-conquer algorithm that, on input integer array a, requires computing the mss of any subarray of

a that contains both $a[n/2-1]$ and $a[n/2]$ (the end of a_{left} and the beginning of a_{right}). For

$a_L \leftarrow \quad \rightarrow a_r$

$$a = -2, 6, -1, 3, -4, 4, -5, 4, -1, 3$$

provide the two arrays S_{LeftSums} and $S_{\text{RightSums}}$ from which the mss (for the middle) can be computed. Based on these two arrays what is the largest mss that includes members of both a_{left} and a_{right}? In general, how many steps does it take to compute this mss, including computing S_{LeftSums} and $S_{\text{RightSums}}$? Explain.

S_{LeftSums}: $-4, -1, -2, \boxed{4}, 2$ $\quad$ $S_{\text{RightSums}}$: $4, -1, 3, 2, \boxed{5}$

$\text{mss}_{\text{middle}} = 4+5 = 9$

$O(n)$ steps to compute the two array sums and to keep track of max member of each array.

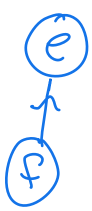
LO3. Solve each of the following problems.

(a) For the weighted graph with edges

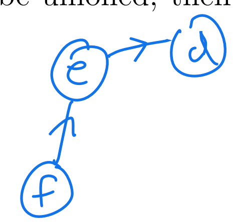
$$(a, e, 6), (b, e, 4), (c, e, 3), (c, d, 5), (d, f, 2), (e, f, 1),$$

Show how the disjoint-set data structure forest changes when processing each edge in Kruskal's sorted list of edges. When unioning two trees, use the convention that the root of the union is the root which has the *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are to be unioned, then the unioned tree should have root a.

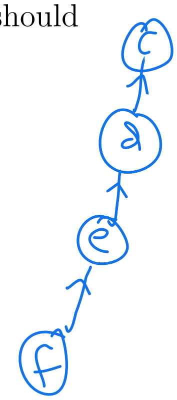
(e, f, 1):



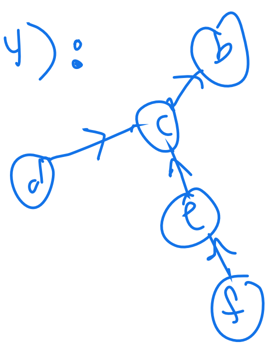
(d, f, 2):



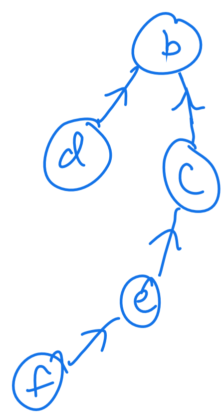
(c, e, 3):



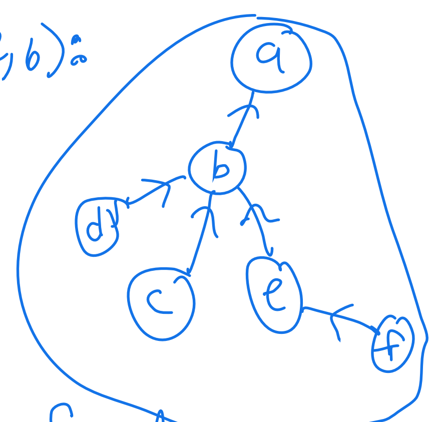
(b, e, 4):



(c, d, 5):



(a, e, 6):



final Tree

(b) The **Fuel Reloading Problem** is the problem of traveling in a vehicle from one point to another, with the goal of minimizing the number of times needed to re-fuel. It is assumed that travel starts at point 0 (the origin) of a number line, and proceeds right to some final integer point $F > 0$. The input includes F, a list of stations $0 < s_1 < s_2 < \cdots < s_n < F$, and a distance d that the vehicle can travel on a full tank of fuel before having to re-fuel. Describe the greedy choice the driver can make in order to minimize the number times that the vehicle has be re-fueled. Using your answer as the basis for a greedy algorithm, for the given instance $d = 7$, $F = 50$, and station locations

$$\{2, 4, 10, 13, 17, 19, 25, 26, 30, 37, 43, 46\},$$

determine the stations that the greedy driver would use to re-fuel?

Drive to the furthest station that is $\leq d$ distance from current location (start, or some earlier station).

Optimal Set: $\{4, 10, 17, 19, 26, 30, 37, 43\}$