CECS 528, Learning Outcome Assessment 5, 10/9/2024, 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 $T(n) = 7T(n/2) + n^3$. Defend your answer.

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

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

then $T(n) = \Omega(n^2)$.

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

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

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.

$$5(\lfloor \frac{1}{2} \lceil \frac{n}{9} \rceil \rfloor - 2) \geq 5(\frac{1}{2} \cdot \frac{n}{9} - 3) = \frac{5n}{18} - 15.$$

$3 \Rightarrow 5$ since now every group of 9 has 5 members that are $\geq$ (respectively $\leq$) the pivot. $5 \Rightarrow 9$ since there are now $\lceil \frac{n}{9} \rceil$ groups.

(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

$$a = \boxed{-5, 8, 3, -9, -3}\boxed{4, 5, -7, 2, -1}$$

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 the middle mss, including computing S_{LeftSums} and $S_{\text{RightSums}}$? Explain.

$S_{\text{LeftSums}} = -3, -12, -9, -1, -6$

$S_{\text{RightSums}} = 4, 9, 2, 4, 3$

Largest mss that overlaps with both a_{left} and a_{Right} is $-1 + 9 = 8$

middle mss is computed in $O(n)$ steps, since there are n additions.

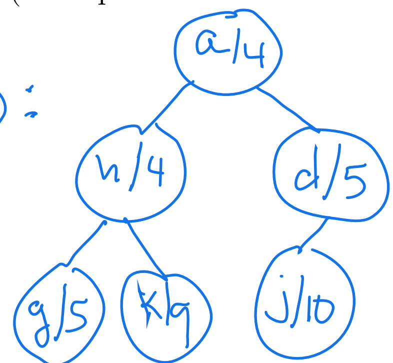
LO3. Solve each of the following problems.

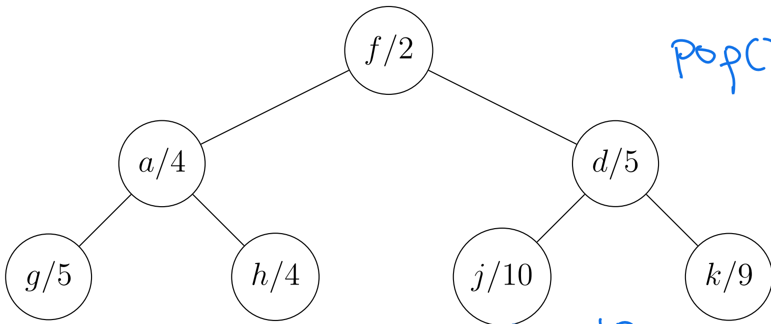
(a) The tree T below shows the state of the binary min-heap at the beginning of some round of Dijkstra's algorithm, applied to some weighted directed graph G. If G has directed edges

$$(f, a, 3), (h, f, 1), (f, j, 3), (f, h, 2), (f, k, 4),$$

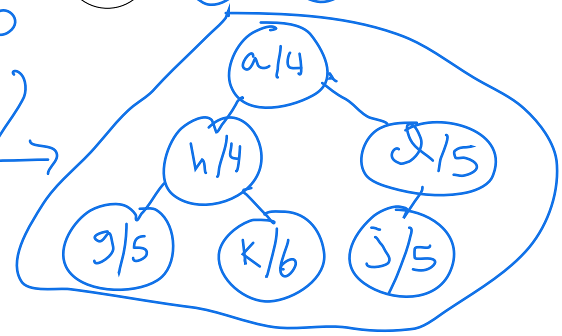
then a plausible state of the heap at the end of the round (hint: provide trees for both the `pop` and `increase priority` operations).

Pop():





Increase Priorities: $j \Rightarrow 2+3=5 < 10$
$K \Rightarrow 2+4=6 < 9$



(b) An instance of **Fractional Knapsack** consists of the following items

item	weight	profit
1	3	40
2	1	50
3	5	70
4	5	50
5	2	40

Greedy Choice: the next item to place in knapsack is that with highest profit-to-weight density

and a knapsack capacity of $M = 10$. State the greedy choice that is made in each round of the greedy fractional knapsack algorithm. Show the sequence of items that are added to the knapsack, including how much (in weight) of each item is added, along with the gain in profit.

Round	Item Choice	Added weight	added profit
1	2	1	50
2	5	2	40
3	3	5	70
4	1	2	$40(2/3) = 26.6$

LO4. 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).

See Lecture Notes

(a) Provide the dynamic-programming recurrence for $d(i, j)$.

See Lecture Notes

(b) Apply the recurrence from Part b to the words u = cbbcca and v = bccaba. Show the matrix of subproblem solutions and use it to provide an optimal sequence of left-to-right edits.

	λ	b	c	c	a	b	a
λ	0	1	2	3	4	5	6
c	1	1	1	2	3	4	5
b	2	1	2	2	3	3	4
b	3	2	2	3	3	3	4
c	4	3	2	2	3	4	4
c	5	4	3	2	3	4	5
a	6	5	4	3	2	3	4

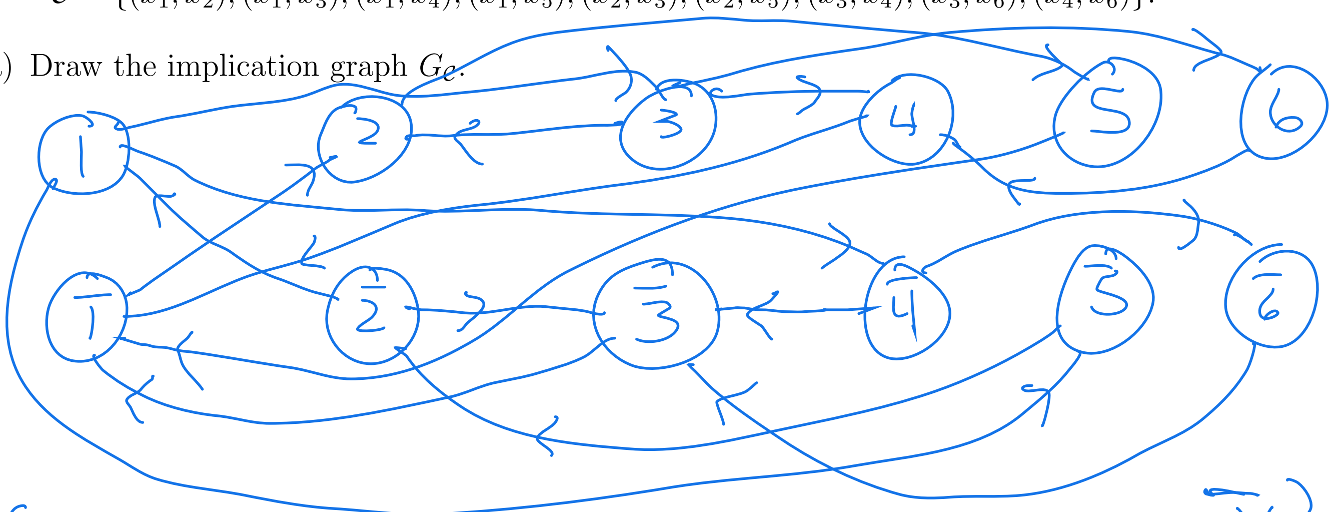
cbbcca
1. +a cbbccaa
2. +b cbbccaba
3. −b cbccaba
4. −c bccaba

~~cb~~bccaba

LO5. Consider the 2SAT instance

$$\mathcal{C} = \{(x_1, x_2), (\overline{x}_1, x_3), (\overline{x}_1, \overline{x}_4), (\overline{x}_1, \overline{x}_5), (x_2, \overline{x}_3), (\overline{x}_2, x_5), (\overline{x}_3, x_4), (\overline{x}_3, x_6), (x_4, \overline{x}_6)\}.$$

(a) Draw the implication graph $G_{\mathcal{C}}$.



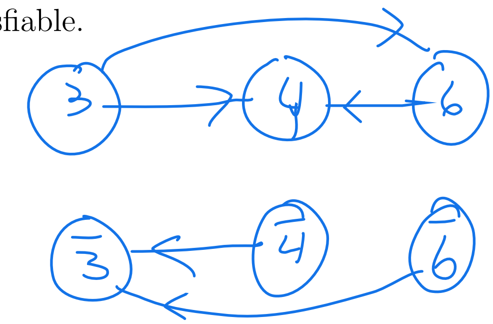
b) $R_{x_1} = \{x_1, \overline{x}_5, \overline{x}_2, x_3, x_2, \overline{x}_3, \overline{x}_1, x_5, \ldots \overline{x}_4, x_4, x_6, \overline{x}_6\}$ is inconsistent

$R_{\overline{x}_1} = \{\overline{x}_1, x_2, x_5\}$ is consistent $\alpha_{R_{\overline{x}_1}} = (x_1 = 0, x_2 = 1, x_5 = 1)$

(b) Perform the `Improved 2SAT` algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced `2SAT` instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable.

b) continued

Reduced Graph: $G_{\mathcal{C}}'$:



$R_{x_3} = \{x_3, x_4, x_6\}$ is consistent

$\alpha_{R_{x_3}} = (x_3 = 1, x_4 = 1, x_6 = 1)$

$\therefore \alpha = (x_1 = 0, x_2 = x_3 = x_4 = x_5 = x_6 = 1)$ satisfies $\mathcal{C}$.

(c) An instance $\mathcal{C}$ of `2SAT` has 512 variables and 2387 different clauses. For the original `2SAT` algorithm, at most many queries will have to be made to the `Reachability`-oracle before one can conclude that $\mathcal{C}$ is satisfiable? Explain.

$1024 = (512)(2)$ maximum since, for all variables x, reachable$(x, \bar{x})$ might be true while reachable$(\bar{x}, x)$ is false.