CECS 528, Exam 1, Spring 2026 Solutions, Dr. Ebert

Directions: Show all necessary work to receive full credit. Please number/letter your problem solutions and make sure your name is on each solution page. You may use both front and back of a page.

Unit 1 LO Problems (25 pts each)

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 9T(n/3) + n^{\log_2 6}$. Defend your answer. (10 pts)

Solution. Since $n^{\log_2 6} > n^{\log_3 9+\varepsilon}$ for any positive $\varepsilon \leq \log_2 6 - 2$, by Case 3 of the Master Theorem, we have $T(n) = \Theta(n^{\log_2 6})$.

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

$$T(n) = T(n/2) + \frac{n}{\log n},$$

then $T(n) = \mathrm{O}(\frac{n}{\log n})$. (15 pts)

Solution. Inductive assumption. $T(k) \leq \frac{ck}{\log k}$ for some constant $c > 0$ and all $k < n$.

Show $T(n) \leq \frac{cn}{\log n}$.

$$T(n) = T(n/2) + \frac{n}{\log n} \leq \frac{c\left(\frac{n}{2}\right)}{\log\left(\frac{n}{2}\right)} + \frac{n}{\log n} =$$

$$\frac{cn}{2(\log n - 1)} + \frac{n}{\log n} \leq \frac{cn}{\log n} \iff$$

$$\frac{c \log n}{2(\log n - 1)} + 1 \leq c \iff \frac{c}{2 - \frac{1}{\log n}} + 1 \leq c \iff$$

$$c\left(1 - \frac{1}{2 - \frac{1}{\log n}}\right) \geq 1 \iff c \geq \frac{1}{\left(1 - \frac{1}{2 - \frac{1}{\log n}}\right)}$$

$\therefore c > 2$ suffices for n

suff. large

LO2. Solve the following problems.

(a) For the `Find Statistic` algorithm, describe **in just a few sentences** how the pivot M is obtained for the partitioning step of the algorithm. Also, the inequality

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

was provided to establish that, for the partitioning step, about 30% of the array elements are less than or equal (respectively, greater than or equal) to M. Explain the significance of each of the following constants that make up the left expression: i) $\frac{1}{5}$ ii) $\frac{1}{2}$, iii) 3. Provide a few sentences for each one. (16 pts)

Solution. The pivot M is selected as the median of the medians of the groups of five that partition the members of array a. i) $\frac{1}{5}$: there are $\lfloor \frac{1}{5} \cdot n \rfloor$ groups, ii) $\frac{1}{2}$: one half the groups have a median that is less than or equal to M, iii) 3: for each group that has a median less than or equal to M, there are a total of at least 3 members in the group that are less than or equal to the pivot.

(b) Consider Karatsuba's algorithm which we'll call `multiply` for multiplying two n-bit binary numbers x and y, where we assume that n is even. Let x_L and x_R be the leftmost $n/2$ and rightmost $n/2$ bits of x respectively. Define y_L and y_R similarly. Let P_1 be the result of calling `multiply` on inputs x_L and y_L, P_2 be the result of calling `multiply` on inputs x_R and y_R, and P_3 the result of calling `multiply` on inputs $x_L + x_R$ and $y_L + y_R$. Then return the value $P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{\frac{n}{2}} + P_2$. Demonstrate the algorithm (only at the root level of the recursion tree!) with $x = 37$ and $y = 11$. Hint: convert x and y to two binary numbers, each having the same (even) number of bits. (9 pts)

Solution. $x = (37)_2 = \underbrace{100}_{x_L=4}\underbrace{101}_{x_R=5}$

$(11)_2 = \underbrace{001}_{y_L=1}\underbrace{011}_{y_R=3}$

$P_1 = (4)(1) = 4 \quad P_2 = (5)(3) = 15 \quad P_3 = (4+5)(1+3) = 36$

$(P_1)(64) + (P_3 - P_1 - P_2)(8) + 15 = 256 + (17)(8) + 15 =$

$256 + 136 + 15 = 407$

LO3. Do the following.

(a) If $p(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ is an unknown polynomial of degree $n-1$, and $\mathrm{DFT}_n(p(x)) = (y_0, y_1, \ldots, y_{n-1})$, then how does one compute the coefficients $(c_0, c_1, \ldots, c_{n-1})$ of $p(x)$? (13 pts)

Solution. $(c_0, c_1, \ldots, c_{n-1}) = \mathrm{DFT}_n^{-1}(y_0, y_1, \ldots, y_{n-1})$.

(b) If Compute $\mathrm{DFT}^{-1}(2, 3+i, 0, 3-i)$ using the IFFT algorithm. Show the entire recursion tree as was done in the lecture notes. (12 pts)

Solution.

$DFT_4^{-1}(2, 3+i, 0, 3-i) =$

$\frac{1}{2}((1,1,1,1) + (1,-i,-1,i) \odot (3,i,3,i)) = (2,1,-1,0)$

$DFT_2^{-1}(2,0) =$

$\frac{1}{2}((2,2) + (1,-1) \odot (0,0)) = (1,1)$

$DFT_2^{-1}(3+i, 3-i) =$

$\frac{1}{2}((3+i, 3+i) + (1,-1) \odot (3-i, 3-i))$

$= (3, i)$

$DFT_1^{-1}(2) = 2$ $DFT_1^{-1}(0) = 0$

$DFT_1^{-1}(3+i) = 3+i$

$DFT_1^{-1}(3-i) = 3-i$

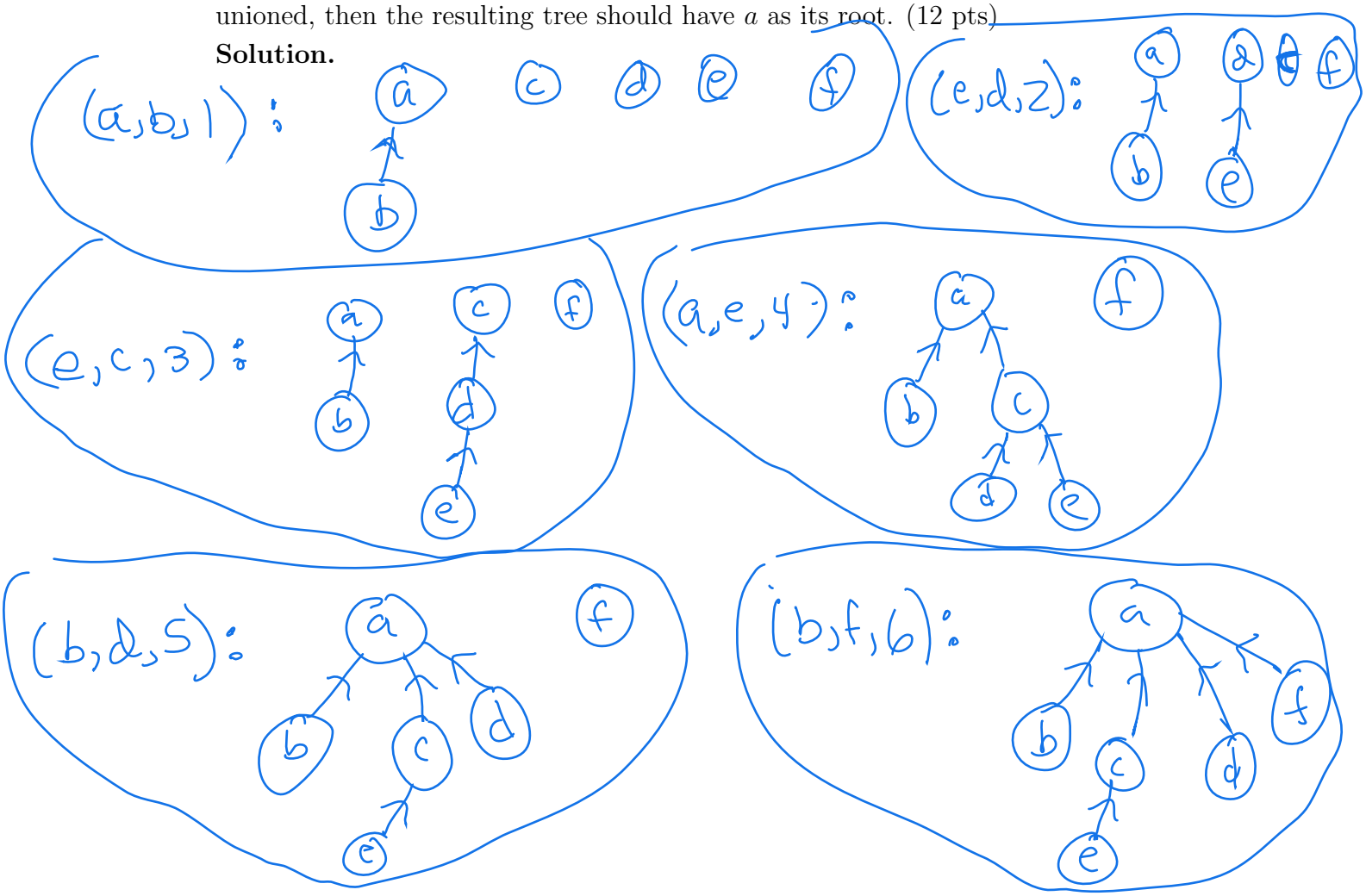
LO4. Do the following.

(a) For the weighted graph with edges

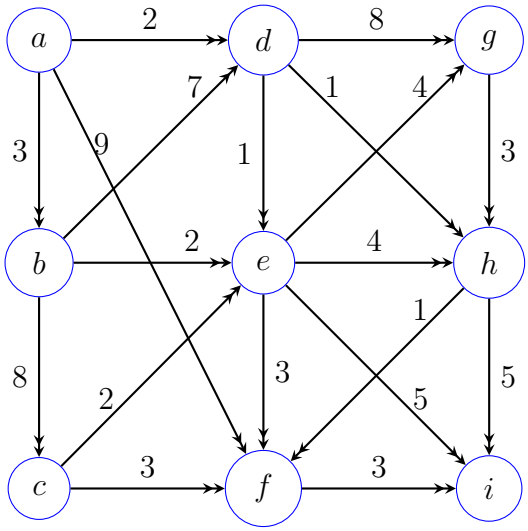
$$(b, d, 5), (a, e, 4), (a, b, 1), (e, c, 3), (b, f, 6), (e, d, 2),$$

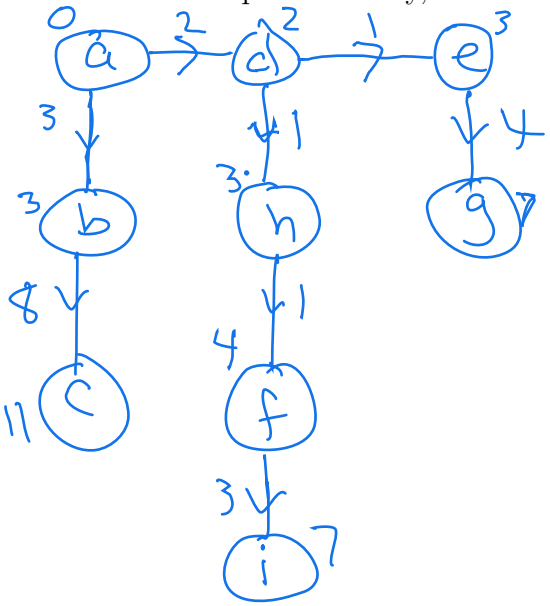
show how the MTree forest changes when processing each edge in the Kruskal sorted order when performing Kruskal's algorithm. When unioning the roots of two trees, use the convention that the root of the resulting tree should be the root having *lower* alphabetical order. For example, if theh roots two trees, one with root a and the other with root b, are unioned, then the resulting tree should have a as its root. (12 pts)

Solution.



(b) For the network (directed, weighted graph) shown below with source vertex a, demonstrate Dijkstra's algorithm by completing the table that shows the vertices that are candidates for being selected in each round, including, for each candidate v, the current estimated distance from a to v, as well as the parent of v in the current best path. Finally, draw the resulting Dijkstra Distance Tree. (13 pts)





Solution.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9
a	0/ϕ	—	—	—	—	—	—	—	—
b	∞/ϕ	3/a	3/a	—	—	—	—	—	—
c	∞/ϕ	∞/ϕ	∞/ϕ	11/b	11/b	11/b	11/b	11/b	11/b
d	∞/ϕ	2/a	—	—	—	—	—	—	—
e	∞/ϕ	∞/ϕ	3/d	3/d	—	—	—	—	—
f	∞/ϕ	9/a	9/a	9/a	6/e	4/h	—	—	—
g	∞/ϕ	∞/ϕ	10/d	10/d	7/e	7/e	7/e	—	—
h	∞/ϕ	∞/ϕ	3/d	3/d	3/d	—	—	—	—
i	∞/ϕ	∞/ϕ	∞/ϕ	∞/ϕ	8/e	8/e	7/f	7/f	—

Additional Problems

A1. Solve the following.

(a) Let a and b be two sorted integer arrays, each having length $n \geq 1$. Provide a pseudocode implementation of the function

```
void merge(int[] a, int[] b, int[] c, int n)
```

that has the side effect of merging a and b into a single sorted array c. You may assume that c has enough memory allocation to handle all $2n$ entries from $a \cup b$. (12 pts)

Solution.

```
Initialize i=0 and j=0. and k=0.
While i<n v j<n,
    If i<n,
        If j<n,
            If a[i] ≤ b[j],
                c[k++] = a[i++].
            Else c[k++] = b[j++].
        Else c[k++] = a[i++]
    Else // j<n
        c[k++] = b[j++]
```

(b) Show how to multiply the complex numbers $a+bi$ and $c+di$ using only three multiplications of real numbers. The algorithm should take a, b, c, and d as input, and produce the real component $ac - bd$ and imaginary component $ad + bc$. Note that the straightforward approach requires four multiplications. We seek a more efficient approach. (13 pts)

Solution.

ac, bc, (a+b)(c-d

A2. Solve the following.

(a) Recall from the lecture notes that F_n is the $n \times n$ matrix for which

$$\mathrm{DFT}(a_0, a_1, \ldots, a_{n-1}) = F_n \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

Provide the entries for F_6^{-1}. Please write all matrix entries in standard form: $a+bi$, using fractions and making use of $\sqrt{3}$ and $\sqrt{2}$ if necessary. (13 pts)

Solution.

$$F_6^{-1} = \frac{1}{6}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2}-\frac{\sqrt{3}}{2}i & -\frac{1}{2}-\frac{\sqrt{3}}{2}i & -1 & -\frac{1}{2}-\frac{\sqrt{3}}{2}i & -\frac{1}{2}+\frac{\sqrt{3}}{2}i \\ 1 & -\frac{1}{2}-\frac{\sqrt{3}}{2}i & -\frac{1}{2}+\frac{\sqrt{3}}{2}i & 1 & -\frac{1}{2}+\frac{\sqrt{3}}{2}i & -\frac{1}{2}-\frac{\sqrt{3}}{2}i \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\frac{1}{2}+\frac{\sqrt{3}}{2}i & -\frac{1}{2}-\frac{\sqrt{3}}{2}i & 1 & -\frac{1}{2}-\frac{\sqrt{3}}{2}i & -\frac{1}{2}+\frac{\sqrt{3}}{2}i \\ 1 & \frac{1}{2}+\frac{\sqrt{3}}{2}i & -\frac{1}{2}+\frac{\sqrt{3}}{2}i & -1 & -\frac{1}{2}+\frac{\sqrt{3}}{2}i & \frac{1}{2}-\frac{\sqrt{3}}{2}i \end{bmatrix}$$

(b) Recall that an instance of the `Task Selection (TSA)` problem is a finite set T of tasks, where each task t is repsented as a triple (id, s, f), where id is the task id, s is when the task needs to start, and $f > s$ is when it needs to finish. The goal is to find a maximum-sized subset M of T for which no two tasks in M overlap. For example, if two tasks in M have respective start and finish times (s_1, f_1) and (s_2, f_2), with $s_1 < s_2$, then it must be true that $f_1 \leq s_2$. The problem can be efficiently solved in a log-linear number of steps using a greedy algorithm. Note: this problem counts for passing part b of LO2.

i. State the greedy choice that is made in order to select a task to add to M in some round of the algorithm. (7 pts)

ii. Using the greedy-choice criterion described in part i, determine a maximum set of nonoverlapping tasks for the following instance of `TSA`.

$$T = \{(1,9,12),(2,11,17),(3,10,12),(4,2,14),(5,2,7),(6,4,9),(7,18,19),(8,5,17),(9,6,17),$$
$$(10,9,20),(11,1,13),(12,9,12),(13,6,15),(14,3,5),(15,16,17)\}.$$

Again each triple represents the id, start time, and finish time. (5 pts)

Solution to i and ii. The greedy choice is to select the next task to be the one that starts at or after the finish time of the previously selected task and (of all such tasks) finishes the earliest. Using this greedy criterion we obtain the optimal set of tasks

$$\{(14,3,5),(1,9,12),(15,16,17),(7,18,19)\}.$$