CECS 528, Exam 2, Fall 2025, Dr. Ebert

IMPORTANT: READ THE FOLLOWING DIRECTIONS. Directions,

- For each problem, write your solution using **ONE SHEET OF PAPER ONLY (BOTH FRONT AND BACK)**. Write **NAME** and **PROBLEM NUMBER** on each sheet.
- Write solutions to different problems on **SEPARATE SHEETS** of paper.
- It is OK to use the same sheet for parts of different make up problems.

Unit 2 LO Problems

LO5. Do the following.

(a) The dynamic-programming algorithm that solves the `Runaway Traveling Salesperson` optimization problem defines a recurrence for the function $\text{mc}(i, A)$. State in words the meaning of $\text{mc}(i, A)$ and provide its recurrence. (7 pts)

(b) Provide the recurrence for the `Optimal Binary Search Tree` optimization problem and use it to compute the minimal weighted access cost for the binary search tree that holds keys 1-3, assuming that keys 1-3 have respective weights 10, 20, and 35. (9 pts)

(c) The following is the dynamic-programming table for an instance of `0-1 Knapsack`. Determine the weights and profits of each item as well as the subset of items that yields an optimal knapsack. (9 pts)

x_i, c	0	1	2	3	4	5	6	7	8
x_1	0	0	0	50	50	50	50	50	50
x_2	0	0	30	50	50	80	80	80	80
x_3	25	30	55	75	80	105	105	105	105
x_4	0	25	45	55	75	95	105	125	125
x_5	0	25	45	55	75	95	105	125	135
x_6	0	25	45	60	80	95	110	130	140

LO6. Draw the implication graph $G_{\mathcal{C}}$ associated with the `2SAT` instance

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

(a) Draw the implication graph $G_{\mathcal{C}}$. (8 pts)

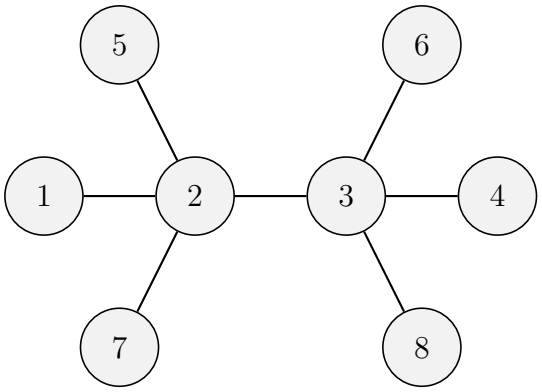
(b) Perform the `Improved 2SAT` algorithm by computing the necessary reachability sets. Use numerical order (in terms of the variable index) and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced `2SAT` instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable. (12 pts)

(c) If an instance $\mathcal{C}$ has 336 variables and 2027 clauses, then what is the least number of queries that must be made to the `Reachability` oracle in order to confirm that $\mathcal{C}$ is satisfiable? Explain. (5 pts)

LO7. Answer the following.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B. (5 pts)

(b) For the mapping reduction f : `Vertex Cover` $\rightarrow$ `Half Vertex Cover`, draw $f(G, k)$ for the `Vertex Cover` instance whose graph is shown below, and for which $k = 2$. (10 pts)



(c) Verify that both (G, k) and $f(G, k)$ are either both positive instances, or are both negative ones. Explain and show work. (10 pts)

LO8. An instance of `Feedback Arc Set (FAS)` is a directed graph $G = (V, E)$ and a natural number $k \geq 0$. The problem is to decide if there is a set S of k vertices of G for which, when removing the vertices of S from G (and all edges incident with them) the resulting graph is acyclic. To see that `FAS` is an `NP` problem we define a certificate for instance (G, k) to be a set $S \subseteq V$ of k vertices. The following pseudocode is used by the verifier to determine if S is in fact a set of vertices whose removal from G creates an acyclic graph. Note: a minimum of 18 points is needed to pass this LO.

$G' = (V', E')$, where $V' = V - S$ and $E' = \{(u, v) \in E \mid u, v \in V'\}$.
For each $u \in V'$,
    For each $v \in V'$,
        If $(u, v) \in E'$ and Reachable(G', v, u), then return 0.
        //v can reach u and so there is a cycle that includes v
Return 1.

(a) Provide size parameters for the `FAS` problem and describe what each represents in relation to an `FAS` problem instance. Hint: there are two of them. (6 pts)

(b) Use the size parameters to provide the big-O number of steps that is required by the verifier to check the validity of a certificate. Hint: a single call to function Reachable requires a linear number of steps with respect to the size parameters of G. Justify your answer. (7 pts)

(c) Classify each of the following problems as being in `P`, `NP`, or co-`NP` (3 points each).

i. An instance of `Fallible` is a Boolean formula F. The problem is to decide if there an assignment that can be made to the variables of F so that F evaluates to 0.

ii. An instance of `Mine Sweep` is a simple graph $G = (V, E, f)$ where $f : V \to \{-1, 0, 1, \ldots\}$ is a function from the set of vertices to the set of natural numbers, including -1. If $f(v) \geq 0$, then it means that a total of $f(v)$ neighbors of v (i.e., vertices that are adjacent to v) must have a mine placed on them. On the other hand, if $f(v) = -1$, then there is no constraint on how many neighbors of v must have a mine. The problem is to decide if there is a function $g : V \to \{0, 1\}$, such that i) $g(v)$ indicates whether or not a mine is placed on vertex v, and ii) for all $v \in V$, if $f(v) \geq 0$, then

$$f(v) = \sum_{u \in N(v)} g(u),$$

where $N(v)$ is the set of all neighobrs of v. In other words, function g meets all the mine constraints that are indicated by f.

iii. An instance of `Palindrome` is an array a of integer, and the problem is to decide if a reads the same forwards as backwards, meaning that, for all $i \in \{0, 1, \ldots, n-1\}$, $a[i] = a[n-1-i]$.

iv. An instance of `Large Vertex Covers` is a graph $G = (V, E)$, and the problem is to decide if every vertex cover of G has size at least $n/2$, where $n = |V|$.

Additional Problems

A1. Consider the set of all words over the alphabet $\{0, 1, 2, 3, 4\}$. For example, $u = 31140$ is one such word. Let $d(u, v)$ denote the **edit distance** between any two such words. Similar to the `Edit Distance` problem, the allowed edits are to add a digit, remove a digit, or change the value of a digit. However, the cost of an edit depends on the value(s) of the digit(s) involved in the edit. For example, adding or removing digit x has a cost equal to x, while changing digit x to digit y incurs a cost equal to $|x - y|$. Again we let $d(i, j)$ denote the edit distance between the i th prefix of u and the j prefix of v.

(a) Based on the rules described above, provide a recurrence for $d(i, j)$. (15 pts)

(b) Apply the recurrence to the words $u = 20243$ and $v = 32440$. Show the matrix of subproblem solutions and use it to identify a "path" that represents an optimal editing sequence. Provide the sequence of changes from right to left. (10 pts)

A2. Do the following

(a) Define what it means to be an instance of i) the `Independent Set IS)` decision problem and ii) the `Vertex Cover (VC)` decision. What exactly is an Independent set in a graph? a vertex cover in a graph? (10 pts)

(b) Describe a valid mapping reduction $f : \text{IS} \to \text{VC}$ from `IS` to `VC`. Defend your answer. (15 pts)

Makeup Problems

LO1. Solve each of the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 4T(n/2) + n^{\log_2 5} \log^3 n$.

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

$$T(n) = 4T(n/2) + 3n,$$

Then $T(n) = \Omega(n^2 \log n)$.

LO2. Solve each of the following problems.

(a) Recall that the `find_statistic` algorithm makes use of Quicksort's partitioning algorithm and uses a pivot that is guaranteed to have at least

$$3(\lfloor \frac{1}{2} \lceil \frac{n}{5} \rceil \rfloor - 2) \geq 3(\frac{1}{2} \cdot \frac{n}{5} - 3) = \frac{3n}{10} - 9 \geq n/4$$

members of a on both its left and right sides, assuming $n \geq 200$. Rewrite all three inequalities/equalities with updated constants, assuming that the algorithm now uses groups of 9 instead of groups of 5. Give the rationale for how you decided to replace the 3 on the left side of the very first inequality.

(b) Consider the following algorithm called `multiply` for multiplying two n-bit binary numbers x and y. In what follows, we assume 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 $P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$. Apply this algorithm to the numbers $x = 13$ and $y = 6$. Only show the top level of the recursion (i.e. do *not* make a recursion tree).

LO3. Do the following.

(a) Consider the FFT algorithm when applied to a polynomial $A(x)$ having degree $2^n - 1$. Provide the equation that relates $A(x)$ to the two subproblem polynomials $A_e(x)$ and $A_o(x)$. What are the degrees of these two polynomials? Based on the equation, we can see the benefit of the 2^n-roots of unity occurring as additive-inverse pairs. What other property of the 2^n-roots of unity is essential for the FFT algorithm? Explain why the property is important.

(b) If $p(x) = -2 + 3x + x^2 - 5x^3$, then compute $\text{DFT}^{-1}(p)$ using the FFT algorithm. Show the entire recursion tree as was done in the lecture notes. (10 points)

LO4. Solve the following.

(a) For the weighted graph with edges

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

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. Hint: remember that the algorithm terminates once a single tree remains.

(b) Recall that an instance of the `Task Selection` problem is a finite set T of tasks, where each task t has a start time $s(t)$ and finish time $f(t)$ that indicate the interval for which the task should be completed by a single processor. The goal is to find a subset T_{opt} of T of maximum size whose tasks are pairwise non-overlapping, meaning that no two tasks in T_{opt} share a common time in which both are being executed. Note: a task with respective start and finish times 2 and 4 does *not* overlap with a task with respective start and finish times 4 and 7, but *does* overlap with a task with respective start and finish times 3 and 6. For the algorithm described in the lecture notes, state the greedy choice that is being made in each step of the algorithm.

Apply the algorithm to the following set of tasks, where each triple in set T represents the id, start time, and finish time.

$$T = \{(1, 90, 120), (2, 110, 170), (3, 100, 120), (4, 20, 140), (5, 20, 70), (6, 40, 90), (7, 180, 190),$$

$$(8, 50, 170), (9, 60, 170), (10, 90, 200), (11, 20, 130), (13, 60, 150), (14, 30, 50), (15, 160, 170)\}.$$

Provide a table that, for each round of the algorithm, shows the value of any statistic that is used to make the greedy choice, and also the task that is selected for that round.