CECS 528, Quiz 3, Friday, Spring 2026, Dr. Ebert

IMPORTANT: READ THE FOLLOWING DIRECTIONS.

- For each LO, write your solutions to all parts using **ONE SHEET OF PAPER ONLY (BOTH FRONT AND BACK)**. Write **NAME** and **PROBLEM NUMBER** on each sheet. Use a moderate font size and section off the page in order to maximize solution space.
- Use **SEPARATE SHEETS** for different LO problems. Please raise your hand if you need more paper.
- **Papers that do not adhere to the above rules will not be graded**.
- The exam ends at 9:00 am

LO9. Solve the following problems.

(a) Recall the mapping reduction $f(\mathcal{C}) = (G, k)$, where f maps an instance of `3SAT` to an instance of the `Clique` decision problem. Given `3SAT` instance

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

answer the following questions about $f(\mathcal{C})$. Hint: to answer these questions you do *not* need to draw G.

i. How many vertices and edges does G have? What is the value of k?. Justify your answer.

ii. $\mathcal{C}$ has the satifying assignment $\alpha = (x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0)$. In relation to mapping reducibility, what does this assignment tell us about G?

(b) Recall the Tseytin transformation f : `SAT` $\rightarrow$ `3SAT` that is used to map reduce an instance F of `SAT` to an instance $f(F)$ of `3SAT`.

i. If $F(x_1, x_2, x_3, x_4) = \overline{x}_4 \wedge (x_3 \vee (\overline{x}_1 \wedge x_2))$, then draw its parse tree and provide the Boolean formula that is satisfiability equivalent to F. and is the starting point of the transformation. Hint: the desired formula has double-arrow subformulas.

ii. Show the steps of the Tseytin transformation that converts the Boolean formula

$$y_1 \leftrightarrow (x_3 \vee y_2)$$

to an intance of `3SAT`. Hint: this formula is unrelated to part i.

LO8. Do the following.

(a) Recall the `Vertex Cover (VC)` decision problem, where an instance consists of a simple graph $G = (V, E)$ and nonnegative integer $k \geq 0$. The problem is to decide if G has a vertex cover of size k. We establish that `VC` is an `NP` problem. A natural certificate for `VC` is set $C \subseteq V$ of k vertices.

i. Provide the pseudocode for a verifier program for `VC` that takes as inputs (G, k) and C and returns 1 iff the C is a k-cover for G's edges.

ii. Given that the size parameters for `VC` are $m = |E|$ and $n = |V|$, determine the big-O number steps required by your verifier. Defend your answer.

(b) Classify each of the following problems as being in `P`, `NP`, or co-`NP`. Note: at least three correct answers is necessary for passing this part of LO8.

i. An instance of `Set Packing` is a positve integer $k > 0$ and a collection of sets $\mathcal{S}$, where each set $C \in S$ is a subset of $\{1, \ldots, n\}$ for some $n \geq 1$. The problem is to decide if, for any k sets $C_1, \ldots, C_k$ in $\mathcal{S}$, their intersection is nonempty.

ii. The `Independent Set` decision problem.

iii. An unstance of `Fallacy` is a Boolean formula F, and the problem is to decide if F always evaluates to 0 regardless of truth assignment to its variables.

iv. An instance of the `Even Cycle` decision problem is a simple graph $G = (V, E)$ and the problem is to decide if G has at least one cycle of even length.

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.

(b) For the mapping reduction f : `Subset Sum` $\rightarrow$ `Set Partition`, provide $f(S, t)$ for the `Subset Sum` instance with $S = \{8, 9, 12, 15, 22, 27\}$ and $t = 36$.

(c) Verify that the reduction is valid for (S, t) in that both (S, t) and $f(S, t)$ are either both positive or negative instances. Justify your answer.

LO6. Do the following.

(a) Consider the `2SAT` instance

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

i. Draw the implication graph $G_{\mathcal{C}}$.

ii. 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) Given `2SAT` instance $\mathcal{C}$, suppose the reachability set $R(x_2)$ contains both x_1 and $\overline{x}_1$. Explain why $R(x_2)$ must also contain $\overline{x}_2$.

LO5. Do the following.

(a) The dynamic-programming algorithm that solves the `Optimal Binary Search Tree (OBST)` optimization problem defines a recurrence for the function $\text{wac}(i, j)$. State in words the meaning of $\text{wac}(i, j)$ and provide its recurrence.

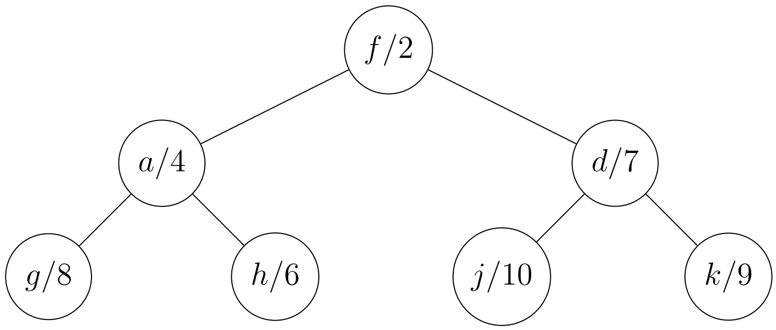
(b) Use the recurrence from part a) to solve the instance of `OBST` that has keys 1,2,3,4 with respective weights 60,30,40,20. Provide the matrix of subproblem solutions (including the value of k associated with each subproblem instance (of size at least two). Draw the optimal binary search tree.

LO4. Do the following.

(a) 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 *directed* graph G. If G has edges

$$(f, d, 1), (a, f, 6), (d, g, 4), (f, g, 3), (f, h, 2), (f, k, 3),$$

then draw a plausible state of the heap at the end of the round.



(b) For the greedy algorithm that solves the **Fuel Reloading Problem**, in *one sentence* describe the greedy choice that is made in each round of the algorithm. Do *not* provide an entire description of the algorithm. Given the station locations

$$10, 13, 19, 23, 26, 32, 36, 47, 56, 66, 73, 77, 89, 98,$$

determine the *least* distance d that a car needs to be able to travel on a full tank of fuel in order to be able to reach location 100 starting from location 0. For this distance, provide a minimum set of stations that it must visit.

LO3. Do the following.

(a) The divide-and-conquer FFT algorithm provides an efficient way to simultaneously evaluate a degree-$(n-1)$ polynomial at n different inputs, where n is a power of two. i) What do we call these n inputs? ii) Provide the equation that relates $A(x)$ to the two subproblem polynomials $A_e(x)$ and $A_o(x)$. iii) Based on your answers to i) and ii), what special property must each of these n inputs possess?

What are the degrees of these two polynomials? Based on this equation, why is it essential that, for even n, the nth roots of unity come in additive-inverse pairs?

(b) If $p(x) = 5 + 3x - 4x^2 - 2x^3$, then compute $\text{DFT}^{-1}(p)$ using the IFFT algorithm. Show the entire recursion tree as was done in the lecture notes.

LO2. Solve each of the following problems.

(a) Provide the arithmetic series that gives the big-O number of steps required by Hoare's `Quicksort` algorithm when the pivot selected is always the second-least element in the array.

(b) Consider Karatsuba's algorithm 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 $P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$. Demonstrate Karatsuba's algorithm on $x = 39$ and $y = 27$, where both number are assumed to require six bits. Limit your demonstration to the root level of recursion. In other words, do not go deeper than level 0.

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 81T(n/3) + n^{\log_2 18}$. Defend your answer.

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

$$T(n) = 2T(n/4) + 5\sqrt{n},$$

then $T(n) = \mathrm{O}(\sqrt{n} \log n)$.