You have 180 minutes to answer six numbered questions.

Write your answers in the separate answer booklet.

Please return this question sheet and your cheat sheet with your answers.

1. For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume *P* ≠ *NP*.** If there is any other ambiguity or uncertainty about an answer, check "No". For example:

   - $x + y = 5$

     Yes [ ] No [X] Suppose $x = 3$ and $y = 4$.

   - 3SAT can be solved in polynomial time.

     Yes [ ] No [X] 3SAT is NP-hard.

   - If P = NP then Jeff is the Queen of England.

     Yes [X] No [ ] The hypothesis is false, so the implication is true.

   Read each statement *very* carefully; some of these are deliberately subtle!

   (a) Which of the following statements are true?

   - The solution to the recurrence $T(n) = 3T(n/3) + O(n^2)$ is $T(n) = O(n^2)$.
   - The solution to the recurrence $T(n) = 9T(n/3) + O(n)$ is $T(n) = O(n^2)$.
   - There is a forest with 374 vertices and 225 edges. (Recall that a *forest* is an undirected graph with no cycles.)
   - Given any directed graph G whose edges have positive weights, we can compute shortest paths from one vertex s to every other vertex of G in $O(VE)$ time using Bellman-Ford.
   - Suppose $A[1..n]$ is an array of integers. Consider the following recursive function:

$$Rizz(i,k) = \begin{cases} 0 & \text{if } i > k \\ 1 & \text{if } i = k \\ \max\left\{ \begin{array}{c} Rizz(i, j-1) + Rizz(j+1, k) \\ + A[i] \cdot A[j] \cdot A[k] \end{array} \,\middle|\, i \le j \le k \right\} & \text{otherwise} \end{cases}$$

     We can compute $Rizz(1, n)$ by memoizing this function into a two-dimensional array $Rizz[1..n, 1..n]$, which we fill by decreasing i in the outer loop and increasing k in the inner loop, in $O(n^2)$ time.

Problem 1 continues onto the next page.

1. *[continued]*

(b) Which of the following statements are true for ***at least one*** language $L \subseteq \{0, 1\}^*$?

- $(L^*)^*$ is finite.
- L is decidable but its complement $\overline{L}$ is undecidable.
- $\{\langle M \rangle \mid M \text{ accepts } L\}$ is undecidable.
- L is the intersection of two NP-hard languages and L is finite.
- There is a polynomial-time reduction from L to the halting problem.

(c) Consider the following pair of languages:

- $\textsc{Tree} = \{G \mid G \text{ is a connected undirected graph with no cycles}\}$
- $\textsc{HamPath} = \{G \mid G \text{ is an undirected graph that contains a Hamiltonian path}\}$

(For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following statements are true, assuming $\text{P} \neq \text{NP}$?

- $\textsc{Tree}$ is NP-hard.
- $\textsc{Tree} \cap \textsc{HamPath}$ is NP-hard.
- $\textsc{Tree} \cup \textsc{HamPath}$ is NP-hard.
- $\textsc{HamPath}$ is undecidable.
- A reduction from $\textsc{Tree}$ to $\textsc{HamPath}$ would imply $\text{P} = \text{NP}$.

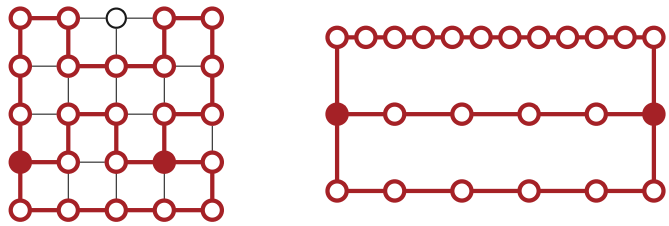
(d) Suppose there is a ***polynomial-time*** reduction R from some language $A \in \{0, 1\}^*$ to some other language $B \in \{0, 1\}^*$. Which if the following statements are ***always*** true, assuming $\text{P} \neq \text{NP}$?

- Problem B is NP-hard.
- If A is finite, then B is finite.
- If A is NP-hard, then B is NP-hard.
- If A is undecidable, then B is undecidable.
- If $A \in P$, then $B \in P$.

Problems 2–6 appear on the next two pages.

2. Submit a solution to ***exactly one*** of the following problems.

   (a) A *theta-graph* is a connected undirected graph in which two vertices have degree 3, and all other vertices have degree 2. Equivalently, a theta-graph is the union of three undirected paths that have the same endpoints, but no other vertices in common. The *size* of a theta-graph is the total number of vertices.

   The 5×5 grid grad contains a theta-subgraph of size 24.

   Prove that is is NP-hard to compute the size of the largest theta-graph that is a subgraph of a given undirected graph G.

   (b) A *clique-partition* of a graph $G = (V, E)$ is a partition of the vertices V into disjoint subsets $V_1 \cup V_2 \cup \cdots \cup V_k$, such that for each index i, every pair of vertices in subset V_i is connected by an edge in G. The *size* of a clique partition is the number of subsets V_i.

   Prove that it is NP-hard to compute the minimum-size clique partition of a given undirected graph G.

   In fact, both of these problems are NP-hard, but we only want a proof for one of them. Don't forget to tell us which problem you've chosen!

3. A *triumph* in a sequence of integers (from the Latin *tri-* meaning "three" and *-umph* meaning "bodacious") is a consecutive triple of sequence elements whose sum is a multiple of 3. For example, the sequence

$$\langle 3, \underline{1, 4, \overline{1,}}\overline{\,5, 9,}\, 6, 2, 3, 5, \underline{8, 9, 7,}\, 9, 3, 2, 3, \underline{8, \overline{4, 6,}}\overline{\,2,}\, 6\rangle$$

   contains five triumphs (indicated by lines above and below).

   We say that one sequence A is *more triumphant* (or *less heinous*) than another sequence B if there are more triumphs in A than in B.

   Describe and analyze an algorithm to compute the number of triumphs in the most triumphant (or equivalently, least heinous) subsequence of a given array $A[1 \,..\, n]$ of integers.

   For example, given the input array $\langle 0, 1, 1, 2, 3, 5, 8, 13, 21\rangle$, your algorithm should return the integer 4, which is the number of triumphs in the most triumphant subsequence $\langle 0, 1, 2, 3, 8, 13, 21\rangle$. Excellent!

Problems 4–6 appear on the next page.

4. Suppose we are given a directed graph $G = (V, E)$, where every edge $e \in E$ has a *positive* weight $w(e)$, along with two vertices s and t.

   (a) Suppose each *vertex* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never visits two consecutive *vertices* with the same color.

   (b) Now suppose each *edge* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never traverses two consecutive *edges* with the same color.

5. let T be a *full* binary tree, meaning that every node has either two children or no children.

   - Recall that the *height* of a vertex v in T is the length of the longest path in T from v down to a leaf. In particular, every leaf of T has height zero.
   - A vertex v is *AVL-balanced* if v is a leaf, or if the heights of v's children differ by at most 1. (You might recall from CS 225 that an ***AVL-tree*** is a binary search tree in which *every* vertex is AVL-balanced.)

   Describe and analyze an algorithm to compute the number of AVL-balanced vertices in T.

6. (a) Let L_a denote the set of all strings $w \in \{0, 1, 2\}^*$ such that $\#(1, w) + 2 \cdot \#(2, w)$ is divisible by 3. For example, L_a contains the strings `0012` and `20210202` and the empty string ε, but L_a does not include the strings `121` or `0122210`.

   Describe a DFA of NFA that accepts L_a. (You do not need to prove that your answer is correct.)

   (b) Let L_b denote the set of all strings $w \in \{0, 1, 2\}^*$ such that no two symbols appear the same number of times, or in other words, the integers $\#(0, w)$ and $\#(1, w)$ and $\#(2, w)$ are all different. For example, L_b contains the strings `110212` and `20220`, but L_b does not include the string `01212` or `2120210` or the empty string ε.

   Prove that L_b is not a regular language.