Write your answers in the separate answer booklet.

You have 180 minutes (after you get the answer booklet) to answer seven questions.

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 $P \neq NP$.** 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 for ***every*** language $L \subseteq \{0,1\}^*$?

   - $(L^*)^*$ is infinite.
   - If L is decidable then its complement $\overline{L}$ is undecidable.
   - $\{\langle M \rangle \mid M \text{ accepts } L\}$ is undecidable.
   - Either L is finite or L is NP-hard.
   - Either L has an infinite fooling set or $L \in P$.

   (b) Consider the following pair of languages:

   - TREE $= \{G \mid G$ is a connected undirected graph with no cycles$\}$
   - HAMPATH $= \{G \mid G$ 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 P $\neq$ NP?

   - TREE is NP-hard.
   - TREE $\cap$ HAMPATH is NP-hard.
   - TREE $\cup$ HAMPATH is NP-hard.
   - HAMPATH is undecidable.
   - A reduction from TREE to HAMPATH would imply P = NP.

Problems 2–7 appear on the next three pages.

2. **More of the same:** Follow exactly the same instructions as problem 1.

(a) Which of the following statements are true?

- The recurrence $T(n) = 3T(n/3) + O(n^2)$ implies $T(n) = O(n^2 \log n)$.
- The recurrence $T(n) = T(n/2) + T(n/3) + T(n/6) + O(n)$ implies $T(n) = O(n)$.
- 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:

$$Ohio(i) = \begin{cases} 0 & \text{if } i < 1 \text{ or } i > n \\ A[i] + \max\{Ohio(i + A[i]),\ Ohio(i - A[i])\} & \text{otherwise} \end{cases}$$

We can compute $Ohio(n)$ by memoizing this function into a two-dimensional array $Ohio[1..n]$, which we fill by increasing i in $O(n)$ time.

(b) Suppose we want to prove that the following language is undecidable.

$$\textsc{Chalmers} := \{\langle M \rangle \mid M \text{ accepts both } \texttt{STEAMED} \text{ and } \texttt{HAMS}\}$$

Professor Skinner suggests a reduction from the standard halting language

$$\textsc{Halt} := \{(\langle M \rangle, w) \mid M \text{ halts on inputs } w\}.$$

Specifically, Professor Skinner claims that if there is a Turing machine SuperNintendo that decides the language Chalmers, then the following algorithm decides Halt.

```
DecideHalt(⟨M⟩, w):
  Encode the following Turing machine:
      AuroraBorealis(x):
        if x = STEAMED or x = HAMS or x = UTICA
            run M on input w
            return False
        else
            return True
  return SuperNintendo(⟨AuroraBorealis⟩)
```

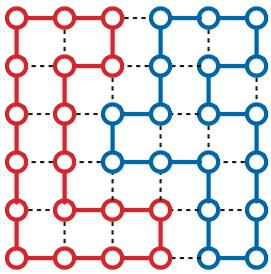
Which of the following statements is true for all inputs $(\langle M \rangle, w)$?

- If M hangs on w, then AuroraBorealis accepts ALBANY.
- If M accepts w, then SuperNintendo accepts $\langle$AuroraBorealis$\rangle$.
- If M hangs on w, then DecideHalt rejects $(\langle M \rangle, w)$.
- DecideHalt decides the language Halt. (That is, Professor Skinner's reduction is correct.)
- We could have proved that Chalmers is undecidable using Rice's theorem instead of this reduction.

Problems 3–7 appear on the next two pages.

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

   (a) A *Hamiltonian bicycle* in a graph G is a pair of simple cycles in G, with identical lengths, such that every vertex of G lies on exactly one of the two cycles.



A Hamiltonian bicycle in the 6×6 grid graph.

   Prove that it is NP-hard to determine whether a given graph G has a Hamiltonian bicycle.

   (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!

4. 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.

5. 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.

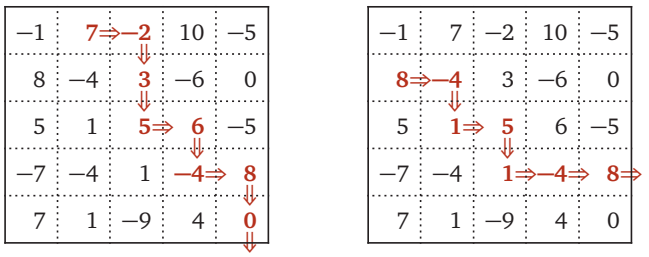
   (a) Suppose each *vertex* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never visits two consecutive *vertices* with the same color.

   (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.

Problems 5 and 6 appear on the next page.

6. *Vankin's Mile* is a solitaire game played on an $n \times n$ square grid. The player starts by placing a token on any square of the grid. Then on each turn, the player moves the token either one square to the right or one square down. The game ends when player moves the token off the edge of the board. Each square of the grid has a numerical value, which could be positive, negative, or zero. The player starts with a score of zero; whenever the token lands on a square, the player adds its value to his score. The object of the game is to score as many points as possible.

   For example, given the grid below, we can score $7-2+3+5+6-4+8+0=23$ points by following the path on the left, or we can score $8-4+1+5+1-4+8=15$ points by following the path on the right.



   (a) Describe and analyze an efficient algorithm to compute the maximum possible score for a game of Vankin's Mile, given the $n \times n$ array of values as input.

   (b) A variant called *Vankin's Niknav* adds an additional constraint to Vankin's Mile: *The sequence of values that the token touches must be a* ***palindrome****.* Thus, the example path on the right is valid, but the example path on the left is not. Describe and analyze an efficient algorithm to compute the maximum possible score for an instance of Vankin's Niknav, given the $n \times n$ array of values as input.

7. (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 strings `01212` or `2120210` or the empty string ε.

   Prove that L_b is not a regular language.