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 ≠ 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 and only if L is infinite.
   - L has a finite fooling set.
   - L is context-free if and only if L is not regular.
   - If there is a Turing machine M that halts on every string in L, then L is decidable.
   - If both L and $\Sigma^* \setminus L$ are acceptable, then L is decidable.

   (b) Which of the following statements are true for ***every*** language $L \subseteq \{0,1\}^*$ and ***every*** language $A \subseteq \{0,1\}^*$ where $A \in NP$?

   - If $L \in$ NP, then $(L \cup A) \in$ NP.
   - If $L \notin$ NP, then $(L \cup A) \notin$ NP.
   - If $L \in$ NP, then $(L \cap A) \in$ NP.
   - If $L \notin$ NP, then $(L \cap A) \notin$ NP.
   - If there is a polynomial time reduction from A to L, then L is NP-hard.

Problems 2–7 appear on the next three pages.

2. **More of the same:** 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".

   (a) Which of the following languages are ***decidable***?

   - $\{\langle M\rangle \mid M$ has at most 374 states$\}$
   - $\{\langle M\rangle \mid M$ accepts a finite number of strings$\}$
   - $\{\langle M\rangle \mid M$ diverges on every input$\}$
   - $\{\langle M\rangle \mid$ There are exactly 374 palindromes that M does *not* accept$\}$
   - $\{\langle M\rangle \mid \langle M\rangle$ is a palindrome$\}$

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

   $$\text{DEMON} := \{\langle M\rangle \mid M \text{ accepts } \texttt{SOULS} \text{ and } \texttt{SHAME} \text{ but diverges on } \texttt{SODAPOP}\}$$

   Your friend Rumi suggests a reduction from the standard halting language

   $$\text{HALT} := \{(\langle M\rangle, w) \mid M \text{ halts on input } w\}.$$

   Specifically, suppose there is a machine TAKEDOWN that decides DEMON. Rumi claims that the following algorithm decides HALT.

```
DecideHalt(⟨M⟩, w):
  Write code for the following algorithm:
    Catchy(x):
      if x = SOULS or x = SHAME
          run M on input w
          ⟨⟨ignore the output of M⟩⟩
          return True
      else
          loop forever
  return TakeDown(⟨Catchy⟩)
```

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

   - If M halts on w, then CATCHY diverges on SODAPOP.
   - If M diverges on w, then TAKEDOWN accepts $\langle$CATCHY$\rangle$.
   - If M halts on w, then TAKEDOWN accepts $\langle$CATCHY$\rangle$.
   - DECIDEHALT decides the language HALT. (That is, Rumi's reduction is correct.)
   - We could instead prove DEMON is undecidable using Rice's theorem.

Problems 3–7 appear on the next two pages.

3. Suppose you are given a ***sorted*** array $A[1..n]$ containing n distinct positive integers, all strictly less than $2n-1$. Describe and analyze an algorithm that finds two *consecutive* integers in this array. Specifically, your algorithm should return an index i such that $A[i]+1=A[i+1]$. To receive full credit, your algorithm must run in sublinear time.

   For example, given the input array $[1, 3, 5, \mathbf{8, 9}, 11, \mathbf{13, 14}, 16]$, your algorithm should return either 4 (because $A[4]=8$ and $A[5]=9$) or 7 (because $A[7]=13$ and $A[8]=14$). *[Hint: Why **must** the array contain two consecutive integers?]*

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

   (a) ***Prove*** that the following **3IN5SAT** problem is NP-hard: Given a boolean formula Φ in conjunctive normal form, with *five* literals in each clause, is there an assignment to the variables such that each clause of Φ contains *at least three* TRUE literals? (The standard 5SAT problem asks for an assignment such that each clause contains at least one TRUE literal.)

   (b) ***Prove*** that the following **67TREE** problem is NP-hard: Given an undirected graph $G=(V,E)$, does G contain a spanning tree T such that every vertex has degree at most 67 in T?

   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! There is a list of (mostly) standard NP-hard problems at the end of the answer booklet.

5. Suppose you are given three strings $A[1..n]$, $B[1..n]$, and $C[1..n]$. Describe and analyze an algorithm to find the length of the longest string that is a subsequence of A, a subsequence of B, and a subsequence of C. For example, given the input strings

   $A=$ `AxxBxxCDxEF`, $B=$ `yyABCDyEyFy`, $C=$ `zAzzBCDzEFz`,

   your algorithm should output the integer 6, which is the length of the longest common subsequence `ABCDEF`.

Problems 6 and 7 appear on the next page.

6. Exactly one of the following languages is regular. Which one?

   (a) $\{\mathtt{0}^a\mathtt{1}^b\mathtt{0}^c \mid a \le b \le c\}$

   (b) $\{\mathtt{0}^a\mathtt{1}^b \mid a \bmod 3 \le b \bmod 3\}$

   Indicate which one of these two languages is regular. Describe a DFA or NFA that accepts the regular language, and ***prove*** that the other language is not regular. (You do not need to prove that your DFA or NFA is correct.)

7. Suppose you are given a directed graph $G = (V, E)$, where each edge has a positive weight, two vertices s and t, and an integer k. Describe an algorithm that computes the length of the shortest walk in G from s to t that traverses ***at most k edges in the wrong direction***. Analyze your algorithm in terms of V, E, and k.

   For example, suppose you are given the graph G shown below.

   - If $k = 2$, your algorithm should return the integer 8, which is the length of the path $s{\rightarrow}u{\leftarrow}v{\rightarrow}w{\rightarrow}x{\leftarrow}t$, which contains two backward edges.
   - If $k = 1$, your algorithm should return the integer 10, which is the length of the path $s{\rightarrow}u{\leftarrow}v{\rightarrow}t$, which contains one backward edge.
   - If $k = 0$, your algorithm should return the integer 19, which is the length of the shortest forward-only path $s{\rightarrow}u{\rightarrow}w{\rightarrow}x{\rightarrow}v{\rightarrow}t$.