CSCI 3104 Fall 2021
Instructors: Profs. Chen and Layer

Problem Set 10

Due Date .... April 26
Name .... **Your Name**
Student ID .... **Your Student ID**
Collaborators .... **List Your Collaborators Here**

Contents

1 Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

2 Standard 26 - Showing problems belong to P

Problem 1. Consider the Shortest Path problem that takes as input a graph $G = (V, E)$ and two vertices $v, t \in V$ and returns the shortest path from v to t. The shortest path decision problem takes as input a graph $G = (V, E)$, two a vertices $v, t \in V$, and a value k, and returns True if there is a path from v to t that is at most k edges and False otherwise. Show that the shortest path decision problem is in P. You are welcome and encouraged to cite algorithms we have previously covered in class, including known facts about their runtime. [**Note:** To gauge the level of detail, we expect your solutions to this problem will be 2-4 sentences. We are not asking you to analyze an algorithm in great detail.]

Answer. We set the edge weights to 1 and apply either the Dijkstra's algorithm or the Bellman-Ford algorithm to find a shortest path, denoted its weight by w. We then return TRUE if $w \leq k$ and FALSE otherwise. The shortest path algorithm takes polynomial time, and verifying if $w \leq k$ takes constant time, so the total running time is polynomial, i.e., the problem is in P. □

3 Standard 27 - Showing problems belong to NP

Problem 2. Consider the Simple Shortest Path decision problem that takes as input a directed graph $G = (V, E)$, a cost function $c(e) \in \mathbb{Z}$ for $e \in E$, and two vertices $v, t \in V$. The problem returns True if there is a simple path from v to t with edge weights that sum to at most k, and False otherwise. Show this problem is in NP.

Answer. To show this problem is in NP we present a poly-time algorithm that verifies a potential solution. A solution to this problem is a path that can be either a sequence of vertices or edges. These are equivalent, so I will assume the answer is a sequence of vertices. First, we need to ensure the sequence is a path from v to t. We traverse the graph starting at v and make sure that each vertex in the the sequence represents a valid edge and that the path ends at t. This can be done in $O(V^2)$ time. Next, we make sure the total cost of the path is less than or equal to k. We do this by scanning the sequence, joining adjacent vertices to get edges, getting the edge weight from $c()$, and comparing the sum to k. Last we check to see if it is a simple path. In $O(V \log V)$ time, we can sort the sequence and look for identical adjacency values. $\square$

Problem 3. Indiana Jones is gathering n artifacts from a tomb, which is about to crumble and needs to fit them into 5 cases. Each case can carry up to W kilograms, where W is fixed. Suppose the weight of artifact i is the positive integer w_i. Indiana Jones needs to decide if he is able to pack all the artifacts. We formalize the Indiana Jones decision problem as follows.

- Instance: The weights of our n items, $w_1, \ldots, w_n > 0$.
- Decision: Is there a way to place the n items into different cases, such that each case is carrying weight at most W?

Show that Indiana Jones $\in$ NP.

Answer. A *certificate* y for Indiana Jones takes the form of a partition of $w_1, \ldots, w_n$ into five sets, $C_1, \ldots, C_5$, such that for each C_i, $W \geq \sum_{x \in C_i} wt(x)$. We can verify y as follows:

1. Verify that all items are present in exactly one C_i and that no items appear more than once. This can be done in $\mathcal{O}(n)$ time.
2. For each C_i, check that $W \geq \sum_{x \in C_i} wt(x)$. This requires a simple sum over the items in each bin, thus taking $\mathcal{O}(n)$ time.

If either (a) or (b) fail to hold, we reject y. Otherwise, we accept it as a valid solution. Since both steps can be computed in linear time, we conclude that Indiana Jones is polynomial-time verifiable and thus belongs to NP. □

4 Standard 27 - NP-compelteness: Reduction

Problem 4. A student has a decision problem L which they know is in the class NP. This student wishes to show that L is NP-complete. They attempt to do so by constructing a polynomial time reduction from L to SAT, a known NP-complete problem. That is, the student attempts to show that $L \leq_p \mathsf{SAT}$. Determine if this student's approach is correct and justify your answer.

Answer. The student's approach is wrong. To prove L is $\mathsf{NP\text{-}Complete}$, we have 2 steps: First, prove L belongs to NP, which is already known. Second, prove a known $\mathsf{NP\text{-}Complete}$ problem can be reduced to L in polynomial time.

Note that as SAT is $\mathsf{NP\text{-}Complete}$, every language in NP reduces **to** SAT in polynomial time. So as $L \in \mathsf{NP}$, we have immediately that $L \leq_p \mathsf{SAT}$. This says nothing about whether L is $\mathsf{NP\text{-}Complete}$. The student would instead need to show that $\mathsf{SAT} \leq_p L$. □

Problem 5. Consider the Simple Shortest Path decision problem that takes as input a directed graph $G = (V, E)$, a cost function $c(e) \in \mathbb{Z}$ for $e \in E$, and two vertices $v, t \in V$. The problem returns True if there is a simple path from v to t with edge weights that sum to at most k, and False otherwise. Show this problem is NP-compelte.

Answer. First we show it is in NP as shown in Problem 2.

ext, we show it is NP-hard by reducing an NP-complete problem to the Simple Shortest Path problem. The most obvious choice here is the Hamiltonian path problem, which is in NP. The Hamiltonian path problem takes an unweighted undirected graph and returns a path that visits every vertex exactly once. The Simple Shortest Path problem takes a weighted directed graph to vertices and returns a path between the two vertices. The reduction starts by converting an undirected graph to a directed graph by making two directed edges of weight -1 for each undirected edge. This takes $O(V)$ time. Using that graph, we call the Simple Shortest Path problem for every pair of vertices in V. If any of those calls return a path that is length $V - 1$ then that path will have visited every vertex once, and it would be a Hamiltonian path.

□