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 can apply Dijkstra's algorithm to G, starting at vertex v. We return an answer of Yes if the path from v to t has weight at most k. Otherwise, we return an answer of No. We note that Dijkstra's algorithm runs in time $O(|E| + |V| \log(|V|))$, which is polynomial. Thus, shortest path problem $\in$ 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. We can apply Dijkstra's algorithm to G, starting at vertex v. We return an answer of Yes if the path from v to t with edge weights that sum to at most k. Otherwise, we return an answer of No. We note that Dijkstra's algorithm runs in time $O(|E| + |V| \log(|V|))$, which is polynomial. Since Dijkstra's algorithm can be solved in polynomial time it can also be checked in polynomial time. Therefore this problem is NP. □

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. We can apply the same algorithm to this problem as we can to the knapsack problem. Using dynamic programming the knapsack problem has a fully polynomial-time answer and is therefore P. Since we use the same algorithm to solve the Indiana Jones problem as the knapsack problem we can say that Indiana Jones $\in P$. Since this problem can be solved in polynomial time it can also be checked in polynomial time. Therefore, Indiana Jones $\in NP$. $\square$

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. To show that a problem is in NP Complete we one have to show that the problem is in NP and two show that a know NP-Complete problem can be reduced to this problem in polynomial time. The students approach is right. They already know that the problem is in NP and they are reducing it in polynomial time to a know NP-complete problem. $\square$

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. We can apply Dijkstra's algorithm to G, starting at vertex v. We return an answer of Yes if the path from v to t with edge weights that sum to at most k. Otherwise, we return an answer of No. We note that Dijkstra's algorithm runs in time $O(|E| + |V|log(|V|))$, which is polynomial. Since Dijkstra's algorithm can be solved in polynomial time it can also be checked in polynomial time. Therefore this problem is NP.

You can reduce a know NP Complete problem in the Hamiltonian Path problem into the shortest path problem in polynomial time.

Since the shortest path problem is in NP and the reduction is true we can say that the Simple Shortest Path problem is NP-complete. □