CSCI 3104 Fall 2021
Instructors: Profs. Chen and Layer

Problem Set 10

where n is input size
$O(n^k)$, and k is a constant.

Due Date .... April 26
Name .... **Your Name**
polynomial time algs:
Student ID .... **Your Student ID**
Collaborators .... **List Your Collaborators Here**

1048.
1049. NP-completeness and the classes P and NP.

Contents

1053. 34.1 polynomial time.

1078. Ch 34.4 1067 Ch 34.3
1068 NP-completeness, NP-completeness and redu...

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. □

1. Apply Bellman Ford to find the shortest path

2. Check if the path is at most k edges.

Step 1 and 2 are both in polynomial time

$\Rightarrow$ It's in P.

A path is simple if all vertices in the path are distinct.

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.

page 1170 of intro to alg.

Answer.

1. Check the given sequence of vertices is a path from v to t $\Rightarrow O(V^2)$

2. Check the total cost of path is $\leq k$

3. Check if the path is a simple path

   $\Rightarrow$ sort the sequence of vertices and look for identical adjacency values.

$O(V \log V)$.

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. □

Given a way to store items into cases.

ex.

$\{ \text{case } 1: w_1, w_3 \}$

$\{ \text{case } 2: W_2, \ldots \}$

$\vdots$

1. Check whethere total weights in each case $\leq W$

$\Rightarrow$ we can do that in $O(n)$

2. Check all items are included in exactly one case.

$\Rightarrow$ $O(n)$

$\Rightarrow$ Indiana Jones $\in$ NP

A

✓

all problems in P $\leq$ NP-Hard $\leq_P$ the problem

B

time complexity

NP-complete

√ 1. NP

√ 2. NP-Hard: Time complexity $\geq$ all problems in P

4 Standard 27 - NP-compelteness: Reduction

Step 1

L P ✓

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 $L \leq_p$ SAT. Determine if this student's approach is correct and justify your answer.

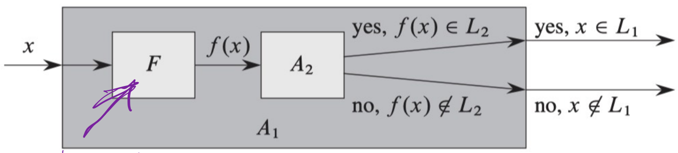
Answer. undirected graph – directed – simple shortest path

Step 2

$L \geq$ all problems in P □

$\geq$ NP-Hard

$L \geq$ NP-Complete : SAT



polynomial time

Figure 34.5 The proof of Lemma 34.3. The algorithm F is a reduction algorithm that computes the reduction function f from L_1 to L_2 in polynomial time, and A_2 is a polynomial-time algorithm that decides L_2. Algorithm A_1 decides whether $x \in L_1$ by using F to transform any input x into $f(x)$ and then using A_2 to decide whether $f(x) \in L_2$.

NP

P

$F + A_2 \leq A_2$

$P + A_2 \leq A_2$

time complexity of solution for L_1

$\leq$ time complexity of solution for L_2.

1) $P = NP$?

2) $P \neq NP$?

x^3

condition 1 $x^c + x^d$ in P in NP

2) $x^c + 2^x = 2^x$ in NP

in NP

2^x not in P

$SAT \leq_p L$

⇒ since $SAT \in NP$, L is at least as hard as NP. In addition L is in NP.

⇒ L is NP-complete.

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.

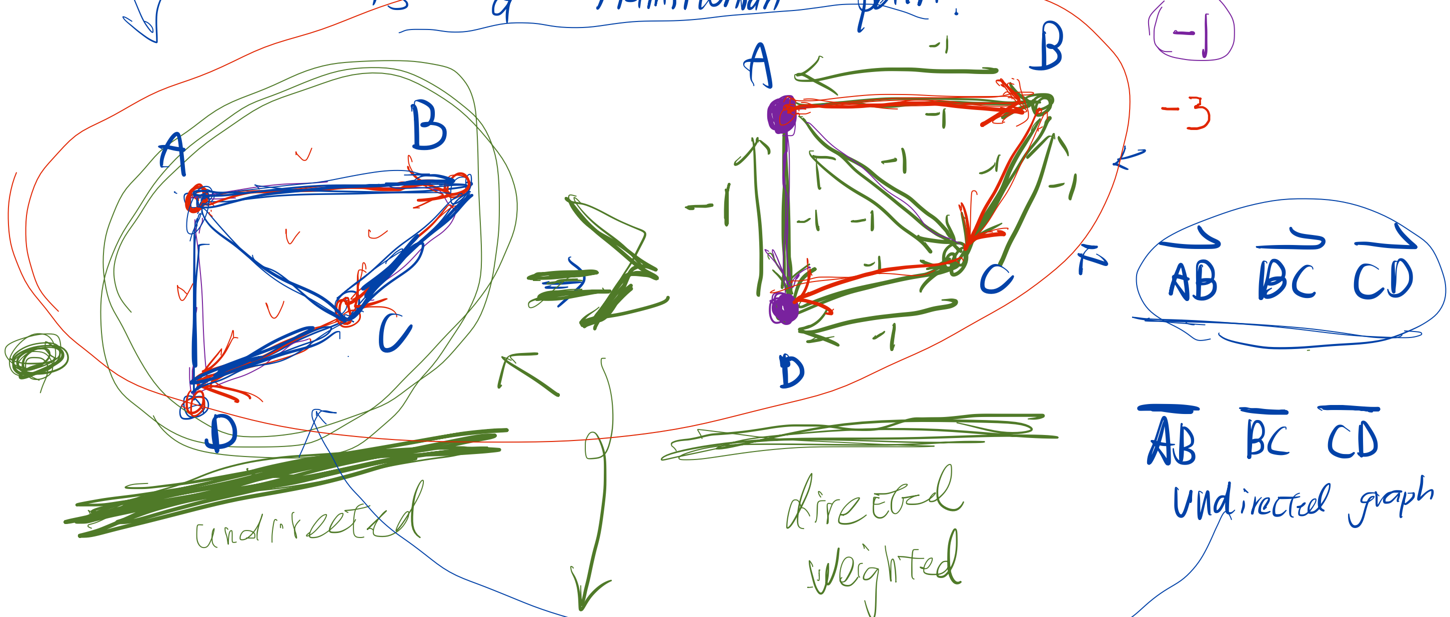
Answer. □

1. Prove SSP is NP as what we did in problem 2

2. Reduce Hamiltonian path problem to SSP. NP-hard

   ⓐ. transform the undirected graph to be a directed graph by making two directed edge with a weight "-1". → $O(V)$

   ⓑ Find the shortest path for each vertex pairs. If any of them return a path with length $|V|-1$. Then it is a Hamiltonian path.



in polynomial time