CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Quiz 19 - DP: Identify subproblems

Due Date ..... April 1
Name ..... **Chengming Li**
Student ID **109251991**

Contents

1 Instructions

- The solutions **should 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 **may not collaborate with other students**. **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, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

2 Standard 19 - Dynamic Programming: Identify Precise Subproblems

Problem 1. Given an undirected graph $G(V, E)$ with positive weight $c_e > 0$ for each edge $e \in E$, you are asked to find the shortest path from a source $s \in V$ to a destination $t \in V$. For each node $v \in V$, denote by d_v the cost of shortest path from v to the destination t. Clearly identify the precise sub-problems to consider. That is, what is the recursive structure to leverage when designing a dynamic programming algorithm?

Answer.

- Recursive structure could be from the node is closest to the destination t to the node is furthest to the destination t.
- At each level or node , we need to consider which edge of current node provide the minimum path cost to the destination t. In other words, we need to consider which edge we should go in order to get the shortest path to next level.
- We need to start the problem by considering the node is closet to the destination which have the shortest path to destination t. At next level, we need to decide which edge of current node provide the shortest path from v to previous node.
- The way to decide it is find the minimum path cost among all the edges of current node v. And the way to decide minimum path is **current** d_v − **previous level**d_v(which is already known as the shortest path to t).

□