

Assignment 7

CSCI 3110: Design and Analysis of Algorithms

Due July 2, 2019

Banner ID: _____ Name: _____

Banner ID: _____ Name: _____

Banner ID: _____ Name: _____

Assignments are due on the due date before class and have to include this cover page. Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

1. (20 pts) The *subset sum* problem is one of the classical NP-hard problems, that is, it is highly unlikely to admit a polynomial-time solution. We will discuss this in more detail in class towards the end of the term. Here's how the problem is defined: Given a set $S = \{x_1, x_2, \dots, x_n\}$ of integers and an integer t , decide whether there exists a subset $S' \subseteq S$ such that $\sum_{x \in S'} x = t$. Note that this means you cannot use the exact same integer twice (but can if there are multiple copies of the same integer, once for each copy).

Even though the subset sum problem is NP-hard, it can be solved under certain assumptions. In particular, subset sum can be solved in polynomial time if $m = \max(S \cup \{t\})$ is at most polynomial in n . This is what you are to prove here: Develop an algorithm that solves the subset sum problem in $O(nm)$ time. Your algorithm should use dynamic programming. As with the dynamic programming examples in class, think of how to build up solutions to subproblems and then store these solutions in a table. If your algorithm claims that there exists such a subset S' , it should output it to prove its claim. Prove that your algorithm is correct and that its running time is as desired.