CSCE629 Analysis of Algorithms

Homework 11

Texas A&M U, Fall 2019 — *11/15/19*

Lecturer: Fang Song — *Due: 10am, 11/22/19*

Instructions.

- Typeset your submission by LaTeX, and submit in PDF format. Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct, and you will get significantly more partial credit if you clearly identify the gap(s) in your solution. You may opt for the "I'll take 15%" option.
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources** for each problem. Be ready to explain your solutions orally to a course staff if asked.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.
- **If you describe a Greedy algorithm, you will get no credit without a formal proof of correctness, even if your algorithm is correct.**

This assignment contains 4 questions, 3 pages for the total of 55 points and 0 bonus points. A random subset of the problems will be graded.

Exercises. Do not turn in.

1. A clique in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E. (In other words, a clique is a *complete* subgraph of G. The size of a clique is the number of vertices it contains. The Clique problems asks to decide whether a clique of a given size k exists in the graph. Show that Clique is NP-complete.
2. Given an integer $m \times n$ matrix A and an integer m-vector b, the 0-1 integerprogramming problem asks whether there exists an integer n-vector x with elements in the set {0,1}suchthat $Ax = b$. Prove that 0-1 integer programming is NP-complete. (Hint: Reduce from 3-CNF-SAT.)

Problems to turn in.

1. (Bonnie and Clyde) Bonnie and Clyde have just robbed a bank. They have a bag of money and want to divide it up. For each of the following scenarios, either give a polynomial-time algorithm, or prove that the problem is NP-complete. The input in each case is a list of the n items in the bag, along with the value of each.

   (a) (10 points) The bag contains n coins, but only 2 different denominations: some coins are worth x dollars, and some are worth y dollars. Bonnie and Clyde wish to divide the money exactly evenly.

   (b) (10 points) The bag contains n coins, with an arbitrary number of different denominations, but each denomination is a nonnegative integer power of 2, i.e., the possible denominations are 1 dollar, 2 dollars, 4 dollars, etc. Bonnie and Clyde wish to divide the money exactly evenly.

   (c) (10 points) The bag contains n checks, which are, in an amazing coincidence, made out to "Bonnie or Clyde." They wish to divide the checks so that they each get the exact same amount of money.

   (d) (10 points) The bag contains n checks as in part (c), but this time Bonnie and Clyde are willing to accept a split in which the difference is no larger than 100 dollars.

2. (15 points) (Galactic Shortest Path) This problem originates from the *Star Wars* movies. Luke, Leia, and friends are trying to make their way from the Death Star back to the hidden Rebel base. We can view the galaxy as an undirected graph $G = (V, E)$, where each node is a star system and an edge (u, v) indicates that one can travel directly between u and v. The Death Star is represented by a node s, the hidden Rebel base by a node t. Certain edges in this graph represent longer distances than others; thus each edge e has an integer length $\ell_e \geq 0$. Also, certain edges represent routes that are more heavily patrolled by evil Imperial spacecraft; so each edge e also has an integer risk $r_e \geq 0$, indicating the expected amount of damage incurred from special-effects-intensive space battles if one traverses this edge.

   It would be safest to travel through the outer rim of the galaxy, from one quiet upstate star system to another; but then one's ship would run out of fuel long before getting to its destination. Alternately, it would be quickest to plunge through the cosmopolitan core of the galaxy; but then there would be far too many Imperial spacecraft to deal with. In general, for any path P from s to t, we can define its total length to be the sum of the lengths of all its edges; and we can define its total risk to be the sum of the risks of all its edges.

   So Luke, Leia, and company are looking at a complex type of shortest path problem in this graph: they need to get from s to t along a path whose total length and total risk are both reasonably small. In concrete terms, we can phrase the *Galactic Shortest-Path Problem* as follows: Given a setup as above, and integer bounds L and R, is there a path from s to t whose total length is at most L, and whose total risk is at most R?

   Describe a poly-time algorithm or prove that Galactic Shortest Path is NP-complete.