CSCE629 Analysis of Algorithms

Homework 7

Texas A&M U, Fall 2019 *10/19/19*

Lecturer: Fang Song *Due: 10am, 10/25/19*

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 3 questions, 4 pages for the total of 60 points and 5 bonus points. A random subset of the problems will be graded.

1. (Burrito-Delivery) You've just accepted a job from Elon Musk, delivering burritos from San Francisco to Houston. You get to drive a Burrito-Delivery Vehicle through Elon's new *Transcontinental Underground Burrito-Delivery Tube*, which runs in a direct line between these two cities. Your Burrito-Delivery Vehicle runs on single-use batteries, which must be replaced after at most 100 miles. The actual fuel is virtually free, but the batteries are expensive and fragile, and therefore must be installed only by official members of the Transcontinental Underground Burrito-Delivery Vehicle Battery-Replacement Technicians' Union. Thus, even if you replace your battery early, you must still pay full price for each new battery to be installed. Moreover, your Vehicle is too small to carry more than one battery at a time.

   There are several fueling stations along the Tube; each station charges a different price for installing a new battery. Before you start your trip, you carefully print the Wikipedia page listing the locations and prices of every fueling station along the Tube. Given this information, how do you decide the best places to stop for fuel?

   More formally, suppose you are given two arrays $D[1,\ldots,n]$ and $C[1,\ldots,n]$, where $D[i]$ is the distance from the start of the Tube to the ith station, and $C[i]$ is the cost to replace your battery at the ith station. Assume that your trip starts and ends at fueling stations (so $D[1] = 0$ and $D[n]$ is the total length of your trip), and that your car starts with an empty battery (so you must install a new battery at station 1).

   (a) (10 points) Describe and analyze a greedy algorithm to find the *minimum number* of refueling stops needed to complete your trip. Don't forget to prove that your algorithm is correct.

   (b) (10 points) But what you really want to minimize is the *total cost* of travel. Show that your greedy algorithm in part (a) does not produce an optimal solution when extended to this setting.

   (c) (5 points (bonus)) Sketch an efficient algorithm to compute the locations of the fuel stations you should stop at to minimize the total cost of travel. You can use any algorithm we've discussed so far.

2. (Minimum spanning tree) This question explores the uniqueness of MST.

   (a) (10 points) Prove or disprove (both directions) that an edge-weighted graph G has a unique minimum spanning tree *if and only if* the following conditions hold:

      - For any partition of the vertices of G into two subsets, the minimum weight edge with one endpoint in each subset is unique.
      - The maximum-weight edge in any cycle of G is unique.

   (b) (10 points) Describe and analyze an algorithm to determine whether or not a graph has a unique minimum spanning tree.

3. Suppose we are maintaining a data structure under a series of n operations. Let $f(i)$ denote the actual running time of the ith operation. For each of the following functions f, determine the resulting amortized cost of a single operation.

   (a) (10 points) $f(i)$ is the largest integer k such that 2^k divides i.

   (b) (10 points) $f(i)$ is i if i is an exact power of 2, and 1 otherwise.