CS1510 Greedy Problems 3, 4 & 5

Rebecca Negley, Sean Myers

September 7, 2011

3. Consider the Change Problem in Austria. The input to this problem is an integer L. The output should be the minimum cardinality collection of coins required to make L shillings of change (that is, you want to use as few coins as possible). In Austria the coins are worth 1, 5, 10, 20, 25, 50 shillings. Assume that you have an unlimited number of coins of each type. Formally prove or disprove that the greedy algorithm (that takes as many coins as possible from the highest denominations) correctly solves the Change Problem. So for example, to make change for 234 shillings, the greedy algorithms would take four 50 shilling coins, one 25 shilling coin, one 5 shilling coin and four 1 shilling coins.

   Solution: Consider trying to make change for 40 shillings. The optimal solution would be two 20 shilling coins. (Observe that this solution must be optimal because there is no 40 shilling coin, so at least two coins are needed.) The greedy algorithm would start by taking the largest denomination possible, namely one 25 shilling coin. It would then need an extra 15 shillings to get 40. In order to fulfill that it would take one 10 shilling coin and one 5 shilling coin. This would make the cardinality of the set of coins from the greedy algorithm 3, while the optimal is 2. Hence, the greedy algorithm does not solve the problem.

4. Consider the Change Problem in Binaryland. The input to this problem is an integer L. The output should be the minimum cardinality collection of coins required to make L nibbles of change (that is, you want to use as few coins as possible). In Binaryland the coins are worth $1, 2, 2^2, 2^3, ..., 2^{1000}$ nibbles. Assume that you have an unlimited number of coins of each type. Prove or disprove that the greedy algorithm (that takes as many coins of the highest value as possible) solves the change problem in Binaryland.

   Solution: Observe that every integer has a unique binary representation. Suppose $L < 2^{1001}$. Then, this unique representation can be thought of as a possible solution to the Change Problem in Binaryland. In every binary position where there is a 1, include one coin with the corresponding value. In every binary position where there is a 0, exclude that coin. If $L \geq 2^{1001}$, include as many 2^{1000}-valued coins as possible, and determine the other coins by the binary representation of $L(mod\ 2^{1000})$ (which will be the remaining value). Observe that this "binary" representation (we use quotes to indicate that there may be more than one of the highest valued coin) is attained by the greedy algorithm. Indeed, it will include as many 2^{1000} coins as possible. After that, it will continue to add on the highest value coin possible. It will only add zero or one of each coin valued lower than 2^{1000}. If it could add two of some coin, it would have instead added one more of the previous coin. Then, the greedy solution contains 0 or 1 of every coin value ($< 2^{1000}$). Hence, it is a "binary" representation of L and, importantly, the same one as above because of uniqueness.

   Suppose the greedy solution differs from the optimal solution. Then, the optimal solution must have 2 or more of some coin (valued $< 2^{1000}$). Otherwise, the optimal solution would be a "binary" representation of L and, by uniqueness, the same one as above. However, if the optimal solution has 2 or more of some coin denomination less than 2^{1000}, this is not optimal because 2 coins representing 2^k could be replaced by one 2^{k+1} coin. Hence the optimal solution must be the"binary" representation and must match the greedy solution.

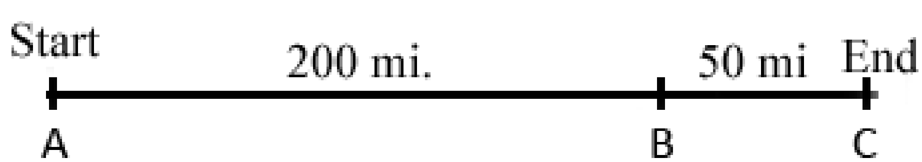
5. You wish to drive from point A to point B along a highway minimizing the time that you are stopped for gas. You are told beforehand the capacity C of your gas tank in liters, your rate F of fuel consumption in liters/kilometer, the rate r in liters/minute at which you can fill your tank at a gas station, and the locations $A = x_1, .., B = x_n$ of the gas stations along the highway. So if you stop to fill your tank from 2 liters to 8 liters, you would have to stop for $6/r$ minutes. Consider the following two algorithms:

   (a) Stop at every gas station and fill the tank with just enough gas to make it to the next gas station.

   Solution: Assume that the greedy algorithm is incorrect. Then, there must be some case I in which the optimal solution is better than the greedy solution. Arrange both the greedy and optimal solutions in order of gas stations $x_1, x_2, \ldots, x_n$. Include every station, inserting a 0 where the car does not actually stop. Going in increasing order, examine the first gas station x_k where the two solutions differ. If the optimal solution stops for less time than the greedy solution at x_k, it will not fill enough gas to make it to the next station. Both solutions are the same to that point, and the greedy solution puts the minimum amount of gas in to make it to the next station. If the optimal solution had put less gas in, it would have therefore been an insufficient amount, and hence it would not be a viable solution. Therefore, the optimal solution must stop for more time than the greedy solution, say l more minutes. Then, we can change the optimal solution so that it stops for the same amount of time as the greedy solution at stop x_k and add l minutes to the optimal solution at stop x_{k+1}. (Note that if x_k had been the last stop, the optimal solution would have filled longer than necessary to make it to the destination and hence not actually been optimal.) The optimal solution then takes the same amount of time and still reaches its destination (since we know it fills enough at x_k to make it to x_{k+1}). Hence, the altered optimal solution is still optimal and agrees with the greedy solution for one more step. Therefore, we can use the exchange argument to reach a contradiction and show that an optimal solution can be found to match the greedy solution.

   (b) Stop if and only if you don't have enough gas to make it to the next gas station, and if you stop, fill the tank up all the way.

   Solution:



Let's say that we have a 10 gallon gas tank with only a fuel economy of 20 miles per gallon and fill at a rate of 2 gallons per minute. In the image above, if we start at point A and have a full gas tank (or fill up our gas all the way), we can make it to point B. Point C is obviously too far away, and so we must refill the car. If we refill the tank 2.5 gallons, this would give an additional 50 miles to get to point C, and we would only wait 2 minutes and 15 seconds. Hence, the optimal solution would take a maximum of 2 minutes and 15 seconds at the gas station. The greedy solution would fill up the entire tank at point C, taking 5 minutes. Therefore the greedy solution must take more time than the optimal solution, so the greedy algorithm is incorrect.