DESIGN AND ANALYSIS OF ALGORITHMS – CSCI 612

NAME: AADITHYA GOPALAKRISHNA BHARADWAJ

ZID: Z1862641

ASSIGNEMENT NUMBER, SEMESTER: Assignment 5, SPRING 2019

==

1. (4 points) Gas stations optimization. There are n gas station S1, …, Sn along I-80 from San Francisco to New York. On a full tank of gas your car goes for D miles. Gas station S1 is in San Francisco, each gas station Si, for 2 ≤ i ≤ n, is di ≤ D miles after the previous gas station Si-1, and gas Sn is in New York. What is the minimum number of gas stops you must make when driving from San Francisco to New York? Give the pseudo-code for a greedy algorithm that solves the above optimization problem. Justify briefly why your algorithm finds the optimum solution. What is the asymptotic running time of your algorithm in term of n?

Answer.)

```
d = 0, k = 1;
for i = 1 to n-1 do
        if d + di+1 > D then d = 0, k = k+1
        d = d +  di+1
return k;
```

Suppose this algorithm returns k gas stops, but the optimum is to refuel at the gas station S_i, S_{i2}, … , S_{im} for m < k. Let $S_{i,j}$ be the first of the gas stations that is later than the j^{th} stop recommended by the algorithm (as such a j must exist because otherwise m ≥ k). Then the car would run out of fuel between S_{ij-1} and S_{ij} , which is contradiction.

Hence it is best to get the fuel as late as possible.

The running time for this algorithm is *O(n)*.

DESIGN AND ANALYSIS OF ALGORITHMS – CSCI 612

2. Consider the problem of making changes for n cents using the fewest number of coins. Assume that each coin's value is an integer.

a. (2 points) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies.

Answer.)

If $n < 5$, use n pennies.

If $5 \leq n < 10$, use 1 nickels and $n - 5$ pennies

If $10 \leq n < 25$ use bn/10c dimes, and then one of the 2 previous cases for $n - 10$bn/10c If $25 \leq n$, use bn/25c quarters, and then one of the 3 previous cases for $n - 25$bn/25c.

Let S_k be the optimal solution when there are k types of changes. when $k = 3$, we have dimes, nickels and pennies for change. To obtain the solution of k+1 types of changes, a greedy choice of using the largest change first will yield optimal solution k+1. This is because for all k+1 types of changes (excluding pennies), they have the greatest common divisor of 5. In other words, an optimal solution can always be obtained by merging small changes to large changes (e.g., one dime = two nickels).

b. (2 points) Suppose that the available coins are in the denominations that are powers of c, i.e., the denominations are c^0 , c^1 , …, c k for some integers c > 1 and k >= 1. Describe a greedy algorithm. (2 bonus points: Argue that the greedy algorithm always yields an optimal solution.)

Answer.)

Let solution S_{k-1} be the optimal when there are c^0 , c^1 , · · · , c^{k-1} denominations. Making a greedy choice for ck will yield optimal solution for S_k. This is because all c^0 , c^1 , · · · , c^k are commonly divisible by c. Thus, for any non-optimal solution, it can be migrate to optimal solution S_k by merging changes from c i to c j , where j > i.

c. (2 points) Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every value of n. You only need to give at least one example.

Answer.)

Let's consider the set of coins { 4 3, 1, }. If we try to make change for 6 cents, the solution from the greedy algorithm would yield $1\times4 + 2\times1 = 6$ (cents). The total amount of coins used in the solution would be $1+ 2 = 3$ (coins). A better solution, however, would be to use $2\times3 = 6$ (cents). The total amount of coins used in this optimal solution would be 2 (coins).