

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.

As a counter-example consider 40 shillings:

$$\text{GREEDY}(40) = 25 + 10 + 5$$

$$\text{OPT}(40) = 20 + 20$$

$\text{GREEDY}(40) \neq \text{OPT}(40)$ therefore GREEDY is not optimal/correct.

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.

Prove \forall input I , $\text{GREEDY}(I) = \text{OPT}(I)$.

Proof: \exists input $I \ni \text{GREEDY}(I) \neq \text{OPT}(I)$.

$$\text{GREEDY}(I) = [g_0, g_1, \dots, g_{1000}]$$

$$\text{OPT}(I) = [o_0, o_1, \dots, o_{1000}]$$

For each denomination smallest to largest, setting OPT to match GREEDY results in an improved OPT, without loss of optimality.

Given $\text{OPT} = [4, 0, 0]$ and $\text{GREEDY} = [0, 0, 1]$

Set $\text{OPT}' = [0, 2, 0]$, the total remains the same: $\text{OPT}' \geq \text{OPT}$.

Set $\text{OPT}'' = [0, 0, 1]$, the total remains the same: $\text{OPT}'' \geq \text{OPT}'$.

Iterate to $\text{OPT} \leq \text{OPT}' \leq \text{OPT}'' \leq \dots = \text{GREEDY}$

Therefore, GREEDY is optimal. \perp

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 $A = x_1, \dots, B = x_n$ 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.

$\text{GREEDY}(I) = [g_0, \dots, g_n]$, where each value represents time spent at each fillup. $\text{OPT}(I) = [o_0, \dots, o_n]$.

Proof: Assume \exists input $I \ni \text{GREEDY}(I) \neq \text{OPT}(I)$

At each step in induction: Add the difference between greedy and optimal to the next element, and change the element in optimal to match Greedy.

Given $G(I) = [5, 6, 7]$ and

$O(I) = [7, 5, 6]$

$O'(I) = [5,7,6]$, the total remains the same: $OPT' \geq OPT$ and driver is guaranteed to reach the next fillup station.

$O''(I) = [5,6,7]$, $OPT'' \geq OPT'$

Iterate to $OPT \leq OPT' \leq OPT'' \leq \dots = GREEDY$

Therefore, GREEDY is optimal. \perp

- (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.

Let $r = 1$, $C = 10$, $F = 1$

Consider three stops at positions 0, 5, and 11

GREEDY: will expend 10 minutes at stop 1 and 5 minutes at stop 2 for a total of 15 minutes.

OPT: will expend 5 minutes at stop 1 and 6 minutes at stop 2 for a total of 11 minutes.

Therefore, GREEDY is not optimal.

For each algorithm either prove or disprove that this algorithm correctly solves the problem. Your proof of correctness must use an exchange argument.