HW #3 - Problem 1

Ryan St.Pierre (ras70)

September 27, 2017

Problem 1A

Before I start the proof I will define a lemma that will be crucial in the eventual proof. The lemma states and proves that given 1,2,5,10 dollar bills for a given bill, b, the value $v \geq b$ cannot be paid optimally/minimally solely with bills smaller than b.

Lemma I

Let the set of possible bills be defined by $s = \{1, 2, 5, 10\}$ and let b_k be the value of the k^{th} bill. That is $b_1 = 1$, $b_2 = 2$, $b_3 = 5$, and $b_4 = 10$.

Lemma I Statement: Given the set of possible bills in s the value Y can not be paid optimally using solely the $1...k-1$ bills if $Y \geq b_k$ and $2 \leq k \leq 4$. In other words, any solution using $1...k-1$ to generate the value Y or greater does **not** use the minimal number of bills (considering all of s).

Lemma I Proof:

This lemma makes a statement about a finite set of k values. Therefore, the lemma will be proved for each value.

k=2

This case is trivial. The lemma states that values of 2 and greater cannot be paid optimally solely using 1 dollar bills. This can be seen by realizing that in any solution two 1 dollar bills can be replaced with one 2 dollar bill and hence made more optimal. In other words a solution cannot be optimal if it includes more than one 1 dollar bill. Thus, the largest optimal solution that can be made using 1 dollar bills given the set s is the value 1. This means that values of 2 and greater cannot be paid optimally with 1 dollar bills, since $1 < 2$. Therefore the lemma statement holds for $k = 2$.

k=3

The lemma in this case (when $k = 3$) states that the set of 1,2 dollar bills cannot be used to pay values greater than 5 optimally. Again only one 1 dollar bill can be used. If more than one 1 dollar bill is used it can be replaced with a single two dollar bill. At most two 2 dollar bills can be used in a minimal bill solution. Three 2 dollar bills can be replaced with a single 1 dollar bill and a 5 dollar bill. These two constraints are given below:

- Number of 1 dollar bills ≤ 1.
- Number of 2 dollar bills ≤ 2.

Given these constraints the largest value that can be paid is 5, where both 2 dollar bills and the single 1 dollar bill are used. However, this value 5 can be generated with a single 5 dollar bill, which is in s. Thus, using solely 1 and 2 dollar bills, no value greater than 4 can be generated optimally. Since $4 < 5$ the lemma holds for $k = 3$.

k=4

The lemma in this case (when $k = 4$) states that the set 1,2,5 cannot be used to pay values greater than 10 optimally when set s exists. The constraints from the $k = 3$ case still holds. In this case there is a further constraint that the number of 5 dollar bills must be less than 2. Any solution with two 5 dollar bills can be made more optimal by replacing the two 5 dollar bills with a 10 dollar bill. The three constraints are given below:

- Number of 1 dollar bills ≤ 1.
- Number of 2 dollar bills ≤ 2.
- Number of 5 dollar bills ≤ 1.

Given these constraints the largest value that can be paid is 10, where both 2 dollar bills, the single 1 dollar bill, and the single 5 dollar bill are used. However, this value 10 can be generated with a single 10 dollar bill, which is in s. Thus, using solely 1, 2, and 5 dollar bills, no value greater than 9 can be generated optimally. Since $9 < 10 = Y$ the lemma holds for $k = 4$.

Having established the lemma and proved its correctness the following proof can be made.

Proof

Assume towards contradiction that an optimal solution OPT exists that is more optimal (uses less bills) than the greedy-pay algorithm (ALG) at paying value x with 1,2,5,10 dollar bills. Let a be the set of n bills used by ALG and b be the set m of bills used by OPT, where m must be less than or equal to n. Given the design of the greedy-pay algorithm a must be in non-increasing order. That is $a[i] \geq a[i+1]$. Let b also be sorted into non-increasing order. This can be done without loss of generality because addition is commutative.

Let k be the index where ALG and OPT first differ in their bill selection. Since ALG selects the largest bill possible possible, if OPT chooses a different bill it must be smaller than that chosen by ALG. That is,

$$a[k] > b[k]$$

Now, let's define two exhaustive cases. *If* k is the last index in a then the values in b cannot possibly equal x. This is true because if k is the last index in a then a and b have the same number of bills. The sum $a[1] + ... + a[k] = x$ and since $a[k] > b[k]$. it must follow that $b[1] + ... + b[k] < x$,. Therefore, in this case OPT cannot pay the full amount and is not a valid solution. This means OPT cannot be an optimal.

If k is *not* the last index in a then we can consider the sum of all elements remaining in a. Since all bills have value greater than or equal to zero it must follow that,

$$a[k] + a[k+1] + ... + a[n] \geq a[k]$$

. It can also be established that all bills $b[k+1]...b[m]$ must be smaller than $a[k]$ since $a[k] > b[k]$ and b is in non-increasing order. Thus, in order for b to be optimal it must also be capable of creating at least a sum of $a[k] + a[k+1] + ... + a[n]$ using bills of size small than $a[k]$. Directly from Lemma I there is no such optimal solution in which b can pay a value greater than or equal to $a[k]$ with bills smaller than $a[k]$. The Lemma can be applied because $b[k] > 1$, since it must be smaller than $a[k]$. Thus, OPT cannot be an optimal solution.

Therefore it has be proven that OPT cannot be optimal and ALG must indeed be the optimal solution.

Problem 1B

Example:

The number needed to pay is 6 dollars and there is one 5 dollar bill and three 2 dollar bills. As stated, unlimited one cent coins can also be used. In other words,

$$x = 6$$

$$\text{set of available bills} = \{\text{one \$5, three \$2, unlimited one cent coins}\}$$

Greedy-Pay Algorithm:

The greedy algorithm will first select the largest possible bill possible, the 5 dollar bill. This would leave 1 dollar left to pay with three 2 dollar bills available. Since no change can be generated the 2 dollar bills can not be used. Thus, at this point one-hundred one cent coins have to be used to pay the final dollar. Therefore the greedy algorithm uses one 5 dollar bill and one-hundred one cent coins, totaling **101 total bills/coins**.

My Better Algorithm:

I will choose to use the three 2 dollar bills to pay the 6 dollars. Thus, my algorithm uses **3 total bills/coins**. Since $3 < 101$ my algorithm is **better** than the greedy-pay algorithm, showing it is not optimal when the differing bills are limited.