York University EECS 3101 March 16, 2018

Homework Assignment #8
Due: March 23, 2018 at 2:30 p.m.

Heili is on her way home from a business trip to Leutonia when she realizes that she forgot to get a souvenir for her friend Tiina. Fortunately, Tiina collects coins, so Heili just goes to the bank in the airport and asks to withdraw an amount of money to get a bunch of coins to give to Tiina. Heili knows that the denominations of Leutonian coins (in klopecks) are $1 = c_1 < c_2 < c_3 < \ldots < c_n$. Moreover, she knows that the teller in the bank will use the standard greedy coin changing algorithm when paying out the money Heili withdraws.

In case you forgot what it is, here is the algorithm that the bank teller follows:

```
Pay(x)
    current ← x // amount still to be paid out
    loop
        exit when current = 0
        q ← max{i : 1 ≤ i ≤ n and c_i ≤ current} // choose the largest possible coin
        give the customer a coin of denomination c_q
        current ← current − c_q
    end loop
end Pay
```

Note that the Leutonian bank teller will use this algorithm even if it does not result in a minimal set of coins. (Leutonian banking laws require this.)

Heili wants to choose an amount x to withdraw so that she gets coins of as many different denominations as possible to give to Tiina. (She only has time to make one withdrawal because she has to catch her plane.)

For example, if the available coin denominations are 1, 4, 6 and 9, Heili can get three different denominations by asking for 14 klopecks $(14 = 9 + 4 + 1)$ or 16 klopecks $(16 = 9 + 6 + 1)$ or 17 klopecks $(17 = 9 + 6 + 1 + 1)$, but she cannot get all four denominations from a single withdrawal. So, the optimal solution would get three different denominations.

While standing in line for a teller, Heili manages to formulate the following lemma.

Lemma: There is an optimal solution in which Heili does not get more than one coin of the same denomination.

Proof sketch: Suppose the optimal number of different denominations that can be attained is d. Let x be the smallest amount that can be withdrawn to obtain d different denominations. Let $y_1 \geq y_2 \geq y_3 \geq \cdots \geq y_m$ be the coins obtained when withdrawing x klopecks. To derive a contradiction, assume $y_i = y_{i+1}$ for some i.

Consider a withdrawal of $x - y_i$ klopecks. When the teller runs the Pay algorithm, the algorithm will pay out coins $y_1, y_2, \ldots y_{i-1}, y_{i+1}, y_{i+2}, \ldots, y_m$. (*Think carefully about why this is true!*) Thus, a withdrawal of $x - y_i$ still provides d different denominations. This contradicts the assumption that x is the *smallest* amount that yields d different denominations. Thus, all the coins that are paid for a withdrawal of x klopecks are distinct denominations. This completes the proof of the lemma.

So from now on, Heili will only think about withdrawing amounts that give her at most one coin of each denomination. Thus, it is just a matter of deciding which set of coin denominations to choose. Then, the amount x that she will ask for will be the sum of those denominations. Having successfully taken EECS3101 in the past, she decides to try a greedy algorithm of the following form.

```
10 ChooseAmount
11     S ← {} // This will keep track of the set of denominations Heili wants to get
12     x = 0 // This will keep track of the sum of the elements of S
13     for i ← 1..n
14         if ________________________ then
15             S ← S ∪ {c_i}
16             x ← x + c_i
17         end if
18     end for
19     output x
20 end ChooseAmount
```

However, Heili is not quite sure how to fill in the test on line 14. (Heili only got a C in EECS3101.)

(a) Fill in the test on line 14 for Heili. Since this is a greedy algorithm, the test should be quite simple.

(b) Prove that the value x computed by ChooseAmount with the test you provided in part (a) always yields the largest possible number of different coin denominations.