#1 Assignment - CMPT 405

Luiz Fernando Peres de Oliveira - 301288301 - lperesde@sfu.ca

September 16, 2018

#1 - Let C be the array containing all the possible coins {*1, 5, 10, 25, 100, 200* }. Let V be the total change value.

Algorithm:

Input: *C, V*
  $d \leftarrow$ sort C such that $d_1 \geq d_2 \geq ... \geq d_n$
  $res \leftarrow \emptyset$; $i \leftarrow 1$
  while $V > 0$ **do**
    if $V \geq d_i$ **then**
      $n_{coins} \leftarrow \left\lfloor \frac{V}{d_i} \right\rfloor$
      $V \leftarrow V - (n_{coins} * d_i)$
      $res \leftarrow res \cup \{(d_i, n_{coins})\}$
    end if
    $i \leftarrow i + 1$
  end while
  return res

Intuition: $\forall (d_i, d_j) \in C$, $1 \leq i < j \leq n$, $d_i \geq 2 * d_j$, meaning that if I the algorithm chooses any d_j over any d_i, it will have to pick at least 2 times more coins for some value V that satisfies both d_i and d_j.

Proof. Now, imagine that the algorithm chooses d_j over d_i, then we have two cases:
- Case 1. $d_i > V$:
  If so, we are done because there are no possible ways of choosing d_i for value V.
- Case 2. $d_i \leq V$:
  If that was the case, we would have an optimal set OPT such that $OPT_{i-1} \cup d_j \subseteq OPT$, which is not the case, once the iteration i will happen before the iteration j, causing the algorithm to choose d_i over d_j (and never the opposite) for any value V that satisfies both d_i and d_j.

The only way to give more coins than the smallest possible number of coins for any change would be in a case where the algorithm chooses d_j over a d_i, which will never happen. $\square$