

Lecture 7: January 25, 2018

Lecturer: Bin Ma

Notes By: Harsh Mistry

7.1 Greedy Algorithms

Greedy algorithm applies to a large class of problems. In general, the solution consists of a series of smaller choices. Since these smaller choices may conflict each other, they need to be optimized together. A brute-forth search will usually take exponential time. Instead, a greedy algorithm uses some simple (and usually heuristic) strategies to make these smaller choices one after another. Once a choice is made, it is committed and won't change. This makes the algorithm very efficient. And one can almost always come up with an efficient greedy algorithm for a problem. However, the optimality of the solution is not always guaranteed. Only in certain cases, the right choice of the greedy strategy may lead to optimal solution of the original problem. And in most of these cases, the algorithm itself is easy. The proof of the optimality is not as easy.

7.1.1 Change Making

A coin system with coin denominations $d_1 < d_2 < \dots < d_n$. A target value V . We want to find a list of coins that add up exactly to V .

One strategy is to repeatedly use the largest coin value that is no greater than the residue target v

Algorithm :

- for i from n to 1
 - output $\lfloor \frac{V}{d_i} \rfloor$
 - $V \leftarrow V - \lfloor \frac{V}{d_i} \rfloor \cdot d_i$

If a coin system is such that the greedy algorithm produces the optimal solution, it's called **canonical**.

Theorem 7.1 *The coin system $2^0, 2^1, \dots, 2^n$ is canonical.*

Proof: We do induction on n . The theorem is correct for $n = 0$. Assume it is correct for $n < n'$. We prove it is also correct for $n = n'$

Case 1 : If $V < 2^n$, then the solution only uses $2^0, 2^1, \dots, 2^{n-1}$. The theorem is true by induction.

Case 2 : if $V \geq 2^n$, suppose the optimal solution uses 2^n exactly k times. We claim that $k = \lfloor V \cdot 2^{-n} \rfloor$. Otherwise, $k < \lfloor V \cdot 2^{-n} \rfloor$. The solution makes up the remaining value $V - k \cdot 2^n \geq 2^n$ with $2^0, 2^1, 2^2, \dots, 2^{n-1}$. By induction, it must use 2^{n-1} at least twice. The two 2^{n-1} can be replaced by 2^n to make the solution smaller. Contradiction.

Now we showed that optimal solution uses 2^n exactly $\lfloor \frac{V}{d_n} \rfloor$ times (The same as the greedy algorithm).

The remaining value is therefore $V - \lfloor V \cdot 2^{-n} \rfloor \cdot 2^n < 2^n$ and only uses $2^0, 2^1, \dots, 2^{n-1}$. By induction, the remaining solution also follows greedy algorithm.

■