CS 341 — Lecture 7

Bartosz Antczak *Instructor: Bin Ma* *September 28, 2017*

7.1 More Greedy Algorithms

This lecture will conclude our topic on greedy algorithms. We will cover a few more examples.

7.1.1 Fractional Backpack

This problem is a sub-problem of the bigger backpack problem.

We have n items each with a respective weight w_i. Every item also has a respective per-unit-weight value v_i (this means that a higher v_i makes is more valuable). The backpack can only hold a max weight of W. We want to find a respective weight for every item x_i such that $\sum_{i=0}^{n} x_i \leq W$ and $\sum_{i=0}^{n} x_i v_i$ is maximized.

Greedy Algorithm

We can assume that the order of all values v_i is sorted. Let's prove that this algorithm is optimal!

```
w ← W
for i = 1 to n do
  if w > 0 then
    x_i ← min{w, w_i}
    w ← w − x_i
  else
    x_i ← 0
```

Proof

Suppose there is an optimal solution that is not a greedy solution, then there exists an $i < j$ such that $v_i > v_j$, $x_i < w_i$ and $x_j > 0$. Now let

$$\Delta = \min\{w_i - x_i, x_j\}$$

But then increasing x_i by Δ while reducing x_j by Δ will only make the solution better. □

7.1.2 Huffman Coding

As taught in CS 240, Huffman coding is a method of encoding letters into binary digits as a means of compressing and decompressing files. In order for this to work, we cannot have any ambiguity! For instance, a poor coding system would be:

$$a = 01 \mid b = 001 \mid c = 011 \mid d = 110 \mid e = 10$$

Observe that 00101110 can be decoded into "bad" *as well as* "bce". A solution to this issue is a **decode tree**. Our Huffman coding problem is, given an input of a frequency table, to find a decode tree that minimizes the average code length.

Greedy Algorithm

1. If there are only two letters, code them with $\{0, 1\}$ and return
2. Let y, z be least frequent letters
3. Replace y, z with new letter w. Let $f(w) = f(y) + f(z)$
4. Construct optimal T' for the new alphabet recursively
5. Let T be w structured as a tree with children y, z and return

Proof

Denote the average code length of a tree T built under an optimal algorithm as $L(T)$. We have

$$L(T) = L(T') + f(y) + f(z) \tag{7.1}$$

$$\ell(w) \cdot f(w) = (\ell(w) + 1)(f(y) + f(z)) \tag{7.2}$$

$$\ell(w) \cdot (f(y) + f(z)) = (\ell(w) + 1)(f(y) + f(z)) \qquad \text{(By definition of our algorithm)} \tag{7.3}$$

$$\tag{7.4}$$

Thus, an optimal T' gives an optimal T.