CS 170 — 2.23 LECTURE 10

PREFIX-FREE CODES / HUFFMAN ENCODING

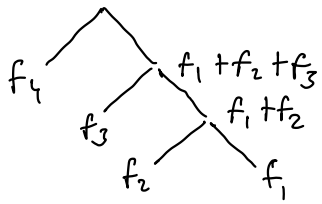
A code is prefix-free if no character is a prefix of another. There is a 1-1 correspondence between such codes and full binary tree — 0 or 2 children, with characters at leaves.

Find a full binary tree with minimal cost, given n = # chars, f_i = frequency of char i. The cost is $\sum_{i=1}^{n} f_i \cdot$ (depth of leaf i, # of bits). Sending m chars requires $m \cdot$ cost bits.

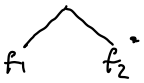
CLAIM Suppose $f_1 \le f_2 \le \dots \le f_n$. Then, f_1 and f_2 are siblings at the bottom of the tree. As in, the greedy algorithm of putting the least frequent char at the bottom & moving up is optimal.

PROOF In lecture notes.

Let's redefine cost(v) for all non-root vertices as $\sum f_i$ of all leaves in subtree rooted at v. Then, the cost of the tree is $\sum$ cost(v) for all non-root vertices.



$\text{total} = 3f_1 + 3f_2 + 2f_3 + f_4$

ALGO Pick 2 least frequent chars $f_1 \le f_2 \le \dots$ and replace them with 1 char $f_1 + f_2$. Find optimal tree with $n-1$ chars. Replace f_1+f_2 with 

PROOF We'll induct on n. If we only 1-2 chars, we use 1-bit encoding. Let T' be the tree found on $f_1+f_2, f_3, \dots, f_n$, which is optimal by induction. Once we modify this to form T, if $v \in T$ and $v \in T'$, then cost(v) is the same. Then, $\text{cost}(T) = \text{cost}(T') + f_1 + f_2$. We know that f_1 and f_2 are siblings at the bottom, so if T is not optimal, we can replace f_1 & f_2 with f_1+f_2 and show that T' isn't optimal. Contradiction.

create Priority Queue $H = \{1, 2, \dots, n\}$ ordered by $f_1 \le f_2 \le \dots \le n$
for $k = n+1$ to $2n-1$:
  i = deletemin(H), j = deletemin(H)
  create node k with children i & j
  $f_k = f_i + f_j$, insert(k)

$O(n \cdot \text{cost of delete}) + O(n \cdot \text{cost of insert}) = O(n \log n)$ binary heap

HORN FORMULAS

Are there boolean values that satisfy $(w \wedge y \wedge z) \wedge (x \wedge y \Rightarrow w) \wedge \ldots$
The general case is NP hard, but we can solve this in limited cases if we only have implications and negative clauses. See Prolog.

1) $(z \wedge w) \Rightarrow w$ 2) $\Rightarrow x$ $(x = T)$ 3) $(\bar{u} \vee \bar{v} \vee \bar{y})$

ALGO Set all variables to false. 1 & 3 are okay, 2 is not.
While an implication is not satisfied, set RHS to T.
If all negative clauses are satisfied, return the assignment.
Else, not satisfiable. Can be done in linear time.

Note: the implications must only be positive.

ENTROPY

Suppose n outcomes with probabilities $p_1, p_2, \ldots, p_n$.

$E(I) = \sum_{i=1}^{n} p_i I(p_i) = \sum_{i=1}^{n} p_i \log_2 \left(\frac{1}{p_i}\right)$ = entropy, measure of randomness.
ranges from 0 to $\log_2 n$.

EE 126 will deal with entropy more. This class will not.
↳ see today's 126 notes.