CS 5112

Algorithms

Huffman Codes

Divide and Conquer

Prefix Codes

Set of characters $S \xrightarrow{\alpha}$ Sequences of $\{0,1\}$
such that if $x, y \in S$, $\alpha(x)$ is not a prefix of $\alpha(y)$

Example $S = \{a, b, c, d, e\}$

a	11
b	01
c	001
d	10
e	000

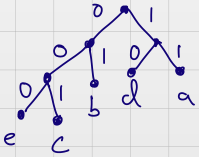
1 1	0 0 1	0 0 0	1 0
a	c	e	d

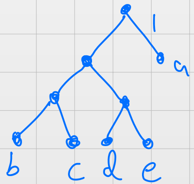
I know this is "a" b/c of the prefix property

Prefix Codes and Binary Trees

α_1

$a \mapsto 11$
$b \mapsto 01$
$c \mapsto 001$
$d \mapsto 10$
$e \mapsto 000$





$a \mapsto 1$
$b \mapsto 000$
$c \to 001$
$d \to 010$
$e \to 011$

Prefix code $\to$ binary tree w/o labels on interior nodes

There is a 1-1 correspondence b/w prefix codes and binary trees with leaves labelled by S.

Huffman Codes

Start w/ two least frequent letters v, w.
Create a new letter $A = v | w$

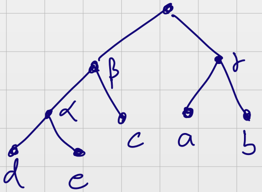
$$f_A = f_v + f_w$$

Recurse on $S' = S \setminus \{v, w\} \cup \{A\}$

Repeat, repeat, repeat, ... until $|S| = 2$.

Huffman Code Example

$f_a = .32$
$f_b = .25$
$f_c = .20$
$f_d = .18$
$f_e = .05$



Step1	Step 2	Step 3
a	a	β
b	b	γ
c	β	
α		

$\alpha = d \mid e$
$f_\alpha = .23$

$\beta = \alpha \mid c$
$f_\beta = .43$

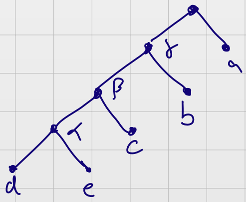
$\gamma = a \mid b$
$f_\gamma = .57$

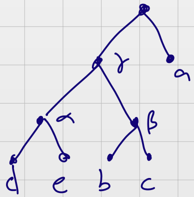
$a \to 10$
$b \to 11$
$c \to 01$
$d \to 000$
$e \to 001$

Huffman Code Example

$f_a = .90$
$f_b = .04$
$f_c = .03$
$f_d = .02$
$f_e = .01$

$f_\alpha = .03$
$f_\beta = .06$
$f_\gamma = .10$





$f_a = .90$
$f_b = .03$
$f_c = .03$
$f_d = .02$
$f_e = .02$

$f_\alpha = .04$
$f_\beta = .06$

Huffman Codes

<u>Huffmann(S,f)</u>

If $|S| = 2$ then
  Let T be tree with one letter set to 0, other 1
Else
  Let v and w be the lowest-frequency letters
  Let $S' = S - \{v, w\} + \{\omega\}$ with:
    $f_\omega' = f_v + f_w$ and $f_x' = f_x$ for $x \in S' - \{\omega\}$
  T' = Huffmann(S',f')
  Let T be prefix tree with leafs v, w added below ω
Return T

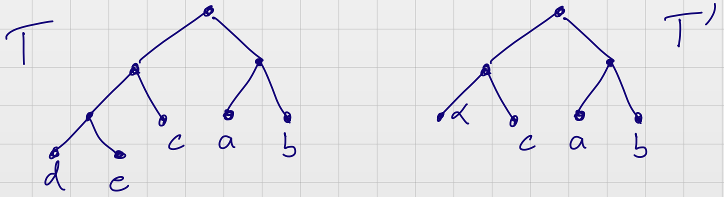
Huffman Codes are Optimal

3 Observations

For an optimal tree T, if $\text{depth}(v) < \text{depth}(w)$, then $f_w < f_v$

There exists an optimal prefix code with the two least frequent characters as siblings.

3rd Observation



$$ABL(T) = ABL(T') + f_\alpha$$

$$= ABL(T') + f_d + f_e$$

Huffman Codes are Optimal

3 Observations

For an optimal tree T, if $depth(v) < depth(w)$, then $f_w < f_v$

There exists an optimal prefix code with the two least frequent characters as siblings.

$ABL(T)$
$= ABL(T') + f_\alpha$

Induction

Induction

(A) $P(k)$ is true $\Rightarrow P(k+1)$ is true
for all $k \geq 1$

Induction

A $P(k)$ is true $\Rightarrow P(k+1)$ is true
for all $k > 1$

If the kth student gets 5 bonus points,
then the $(k+1)^{st}$ student gets 5 bonus points

Induction

A $P(k)$ is true $\Rightarrow P(k+1)$ is true for all $k > 1$

If the kth student gets 5 bonus points, then the $(k+1)^{st}$ student gets 5 bonus points

B $P(1)$ is true

Induction

(A) $P(k)$ is true $\Rightarrow P(k+1)$ is true for all $k > 1$

(B) $P(1)$ is true

If the kth student gets 5 bonus points, then the (k+1)st student gets 5 bonus points

The 1st student gets 5 bonus points

Induction

A $P(k)$ is true $\Rightarrow P(k+1)$ is true for all $k > 1$

If the kth student gets 5 bonus points, then the $(k+1)^{st}$ student gets 5 bonus points

B $P(1)$ is true

The 1^{st} student gets 5 bonus points

If A and B then $P(k)$ is true for all $k \geq 1$

Induction

"inductive step"

A $P(k)$ is true $\Rightarrow P(k+1)$ is true for all $k > 1$

If the kth student gets 5 bonus points, then the $(k+1)^{st}$ student gets 5 bonus points

"base case"

B $P(1)$ is true

The 1^{st} student gets 5 bonus points

If A and B then $P(k)$ is true for all $k \geq 1$

Everybody gets 5 bonus points!

Huffman Codes are Optimal

Proof by induction on $k = |S|$ ($P(k)$ is "HC on S w/ $|S| = k$

Base case: $k = 2$ Optimal tree has two leaves 

is optimal")

Inductive step ($P(k) \Rightarrow P(k+1)$): If Huffman codes are optimal for $|S| = k$, then they are optimal for $|S| = k+1$

Assume HC are optimal for $|S| = k$.

Let T be the tree for a HC for S w/ $|S| = k+1$.

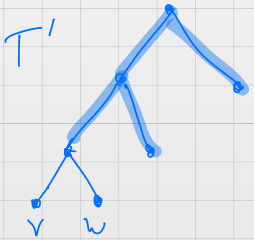
Let Z be an optimal tree for S. By obs 2, can assume the two least frequent chars, v and w, are siblings in Z.

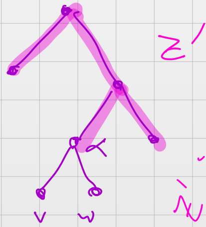
T Huffman Code

$|S| = k+1$

$|S'| = k$

Z Optimal





Z' is Z contracted on $\alpha = v, w$

$$ABL(T) = ABL(T') + f_\alpha \qquad ABL(Z) = ABL(Z') + f_\alpha$$

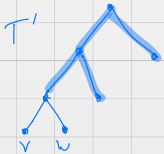
T' must be optimal (inductive hypothesis) $\Rightarrow ABL(T') \leq ABL(Z')$

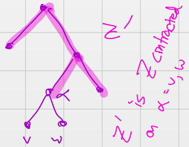
T Huffman Code

$|S| = k+1$

$|S'| = k$

Z Optimal





$$ABL(T) = ABL(T') + f_\alpha \qquad ABL(Z) = ABL(Z') + f_\alpha$$

T' must be optimal (inductive hypothesis) $\Rightarrow ABL(T') \leq ABL(Z')$

$\Rightarrow ABL(T) \leq ABL(Z)$

$\Rightarrow T$ is optimal.

Divide and Conquer: Sorting

Let's start with a greedy approach

Divide and Conquer: Sorting

Let's start with a greedy approach

Bruteforce-Sort(L)

While $|L| > 0$
    Find minimum value x in L
    Append x to L'
    Remove x from L

Return L'

Runs in $O(n^2)$

Divide and Conquer

1. Divide the input into subproblems
2. Solve each subproblem
3. Carefully merge the solutions (often $O(n)$)

Merge Sort

Merge-Sort(L)

If $|L| = 1$ then Return L

Split L into two halves A, B
A <- Merge-Sort(A)
B <- Merge-Sort(B)
L <- Merge(A,B)

Return L

Run time:

$T(n)$:= runtime of MS on a list of length n,

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + cn$$
$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + c\left(\frac{n}{2}\right)$$

Merge Sort

Runtime:

$T(n)$:= runtime of MS on a list of length n,

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + cn$$
$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + c\left(\frac{n}{2}\right)$$

$$2T\left(\frac{n}{2}\right) + cn$$
$$4T\left(\frac{n}{4}\right) + cn$$
$$8T\left(\frac{n}{8}\right) + cn$$
$$\vdots \qquad + cn$$
$$nT(1)$$

$$(\#\text{recursions}) \cdot cn + n$$
$$= \log n \cdot cn + n = O(n \log n)$$