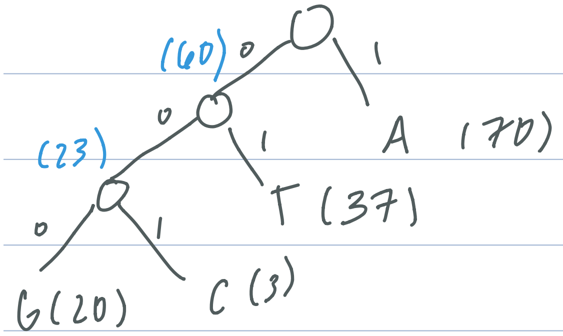
CS124 Lecture 9

Feb 21, 2020

Huffman Encoding Pt. 2



	prefix code	
A - 1		70 x1
T - 01		37 x 2
G - 000		20 x 3
C - 001		3 x 3
		213 million

Decoding sequence = walk on tree, back to root

60+70+37+23+20+3= total cost = 213 million

Greedy Alg

Take 2 least frequent symbols

merge them into a "meta node"

continue

~~A 60~~

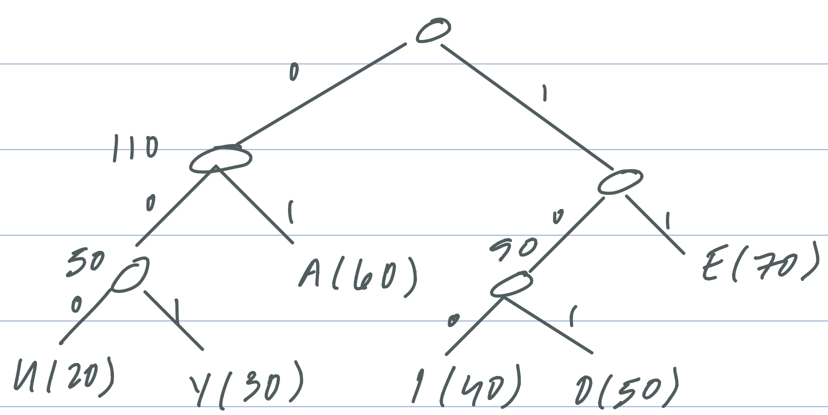
~~E 70~~

I 40

O 50

U 20

Y 30

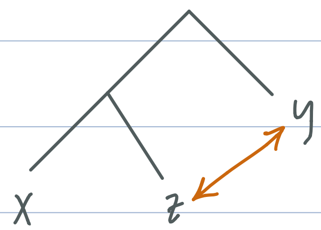


Claim 1. W/o loss of generality, there is an optimal tree that has the two least frequent nodes as siblings at deepest level of the tree.

Pf. By contradiction:

Case 1: Not siblings, but at the same level, swap to make them siblings. No change ✓

Case 2: Not siblings, different levels

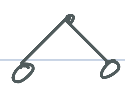


swap = more optimal tree because y is less frequent than z

$\Delta \text{cost} = f_y \cdot l_y + f_z \cdot l_z - (f_y \cdot l_z + f_z \cdot l_y) = (f_y - f_z)(l_y - l_z)$

<u>Huffman coding works by Induction</u>

Base case: 2 nodes, trivial



Inductive step: Suppose true for n steps, true for $n+1$

· First step combines 2 least frequent

optimal tree on n nodes, with these two collapsed.

"expanding" out combined node is optimal tree

- By contradiction
  - Assume better tree on n+1 nodes
  - OPT tree T' has 2 least freq nodes as siblings on bottom
  - T' gives S' on n nodes by collapsing 2 least freq nodes

$cost(S) \leq cost(S')$

$cost(T) = cost(S) + $ Freq of 2

$cost(T') = cost(S') + $ Freq ...

T' is not a better tree $\longrightarrow cost(T) \leq cost(T')$

Divide and Conquer (+ Combine)

ex: mergesort

$T(n) = 2T(n/2) + O(n)$

$\hookrightarrow O(n \log n)$

<u>Finding max/min for n numbers</u>

max n-1 operations, minimum n-1 operations

Split #s in half

Find max/min of each ½

2 additional comparisons to combine

$T(n) = 2T(n/2) + 2$

$T(2) = 1$ $\quad T(8) = 10$ $\quad T(n) = an + b$

$T(4) = 4$ $\quad T(16) = 22$ $\quad a = 3/2, \; b = -2$

Multiplication

multiplying n-digit #s is $O(n^2)$ time by standard method.

$n=8$

$$\begin{array}{r} 6\,3\,4\,1\,8\,7\,1\,8 \\ \times\ 2\,5\,1\,9\,2\,4\,3\,9 \\ \hline \end{array}$$

$x = 10^{n/2} \times a + b$

$y = 10^{n/2} \times c + d$

$x \cdot y = 10^{n}(a \times c) + 10^{n/2}(a \times d + b \times c) + b \times d$

$T(n) = 4 \cdot T(n/2) + O(n)$

↳ by master theorem, $O(n^2)$

$(a+b)(c+d) - ac - bd = ad + bc$

→ $T(n) = 3 \cdot T(n/2) + O(n)$

Matrix Mult

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

$T(n) = 8T(n/2) + O($