

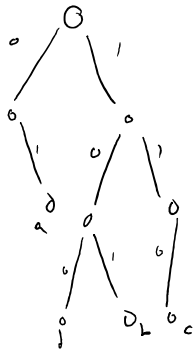
Huffman Codes

$C =$ set of n characters, $f: C \rightarrow \mathbb{N}$ frequency, find binary code $code(c)$ s.t.

- 1) no code is a prefix of another one
- 2) $\sum_{c \in C} f(c) \cdot |code(c)|$ is minimal ($|c|$ is length of code).

binary tree:

a 01
b 101
c 110
d 100



leaves are character codes.

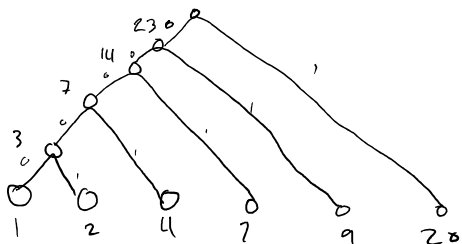
$d(c) =$ depth of leaf corresp. to c .

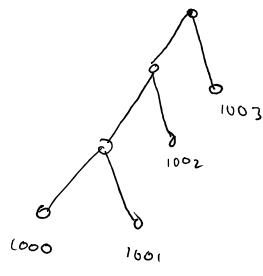
$$\sum_{c \in C} f(c) \cdot |code(c)| = \sum_{c \in C} f(c) d(c)$$

given n nodes, construct binary tree T s.t. $\sum_{x \in \text{leaf}(T)} f(x) \text{depth}(x)$ minimal.

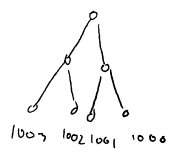
Note: T must be full (every non-leaf has 2 children)

Greedy algorithm.





$\approx 9000 \times$



$\approx 8006 \checkmark$

runtime n^2 if use array.

Use priority queue to get $n \log n$ time.

Priority Q:

insert/delete
in $\log n$ time.

Q contains roots of trees in forest.

This algorithm always generates optimal soln.

write it recursively to prove optimality