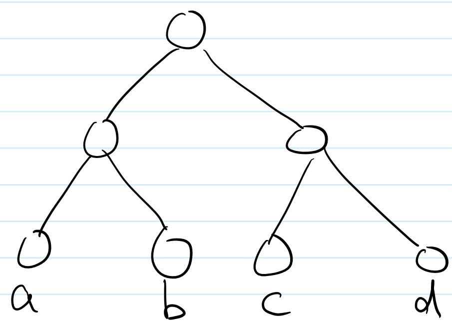
Lecture 7 Greedy Algorithms (cont'd)

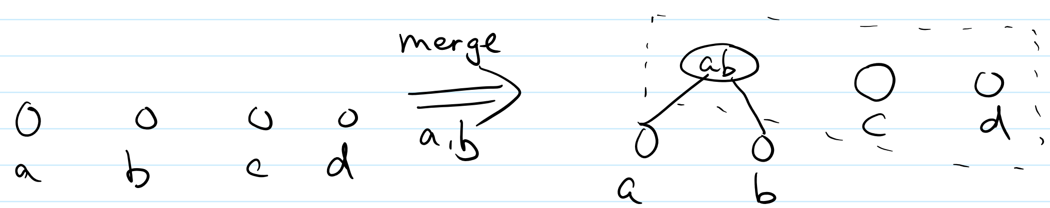
Tuesday, September 19, 2017 2:33 PM

- Hoffman Tree

  a : 5   b : 10   c : 3   d : 6



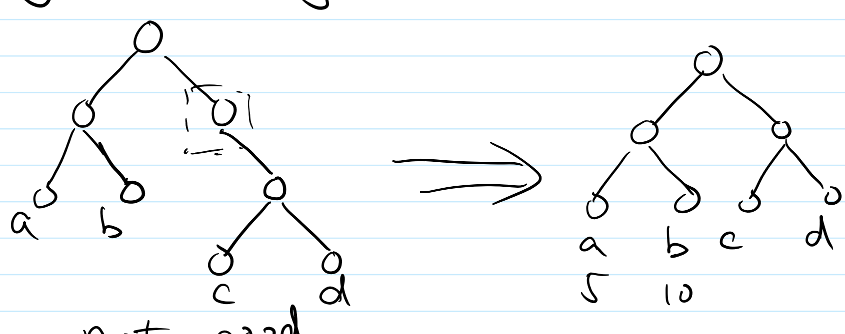
- decision to make?
  - can construct a tree by merging its leaves.



tree with n leaves, n-1 merging operation create a binary tree.

- binary tree is "good" if all intermediate nodes have two children

Claim: any "good" binary tree can be constructed by merging.

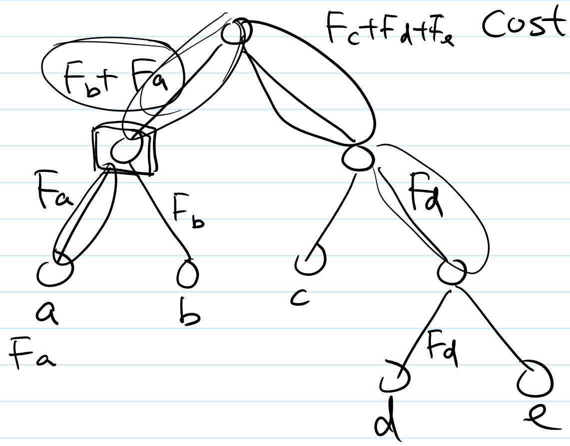


not good

- how to make the greedy choice?
  - need a way to compute "cost" for merging operation.

Claim: Define cost of merging a b to be sum of frequencies of a.b.

sum of costs of the n-1 merging operation = cost of the tree sum of costs of the $n-1$ merging operation = cost of the tree



$\text{Cost} = \sum_{i:\text{leaves}} \text{Freq}_i \times \text{Depth}_i$

$= \sum_{e:\text{edges}}$ [total frequency of characters below this edge]

$= \sum_{e:\text{edges}}$ merge cost when the lower node is created

$+ \left[\sum_{i:\text{leaves}} \text{Freq}_i\right]$

||

final merge cost

---

greedy algorithm: minimize immediate cost

Choose two characters that have lowest frequencies

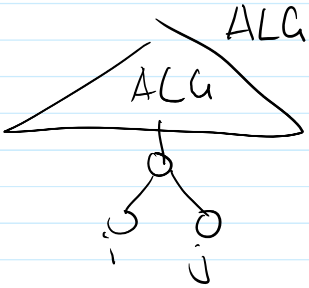
– Proof of correctness:

Use induction: IH: ALG is optimal for all alphabets with n characters.

Base case $n=2$ trivial

induction step: assume IH holds for n, now if the alphabet has $n+1$ characters.

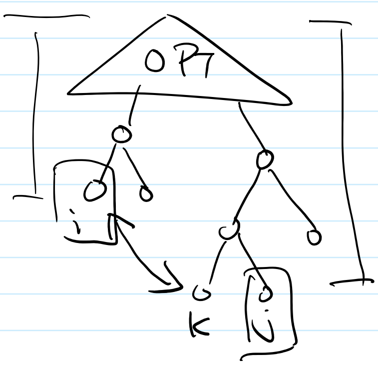
ALG has a solution that merges i, j first (two characters with smallest frequency)



assume towards contradiction that there is a better solution OPT

if in OPT, i and j are not merged

look at node i and j in OPT



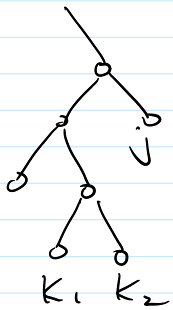
wlog. $\text{depth}_i^{OPT} \leq \text{depth}_j^{OPT}$

case 1 j has a sibling k

swapping i, k cannot increase cost

because $\text{Freq}_i \leq \text{Freq}_k$

case 2 let k_1, k_2 be a pair of leaves that is merged in the sibling tree of i



Case 2 let k_1, k_2 be a pair of leaves that is merged in the sibling tree of j

$$\text{Depth}^{OPT}_{k_1} = \text{Depth}^{OPT}_{k_2} > \text{Depth}^{OPT}_{j}$$

swapping $(i, k_1), (j, k_2)$ will decrease cost

- Can always transform OPT into OPT' where i, j are merged first.
- by induction hypothesis ALG is optimal after merging i, j so ALG is also optimal for this alphabet of size $n+1$ □