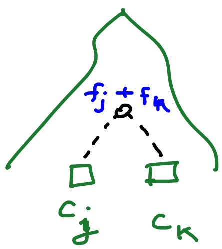
# Huffman Codes — Continued

**Claim:** Let $C = \{(c_i, f_i) \mid 1 \le i \le n\}$ be an instance of the (Huffman coding) problem. Let $f_j$, $f_k$ be two minimum frequencies ($j \ne k$) in $C$. Then there is an optimal solution (to $C$) with choice $f_j, f_k$.

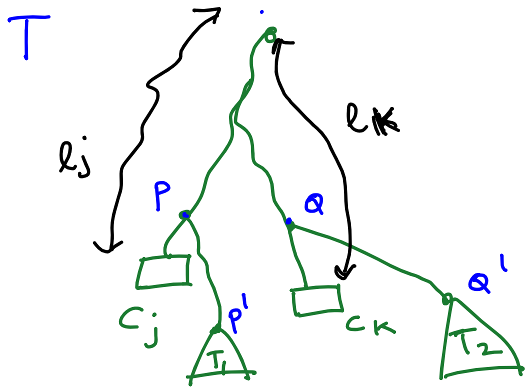
**Proof:** Let $T$ be tree corresponding to codes of some optimal solution of $C$.

$T$




If leaves $c_j$, $c_k$ are children of some internal node $u$, then this optimal solution can be seen as greedy choice + solution to the subproblem.

So we are done in this case.

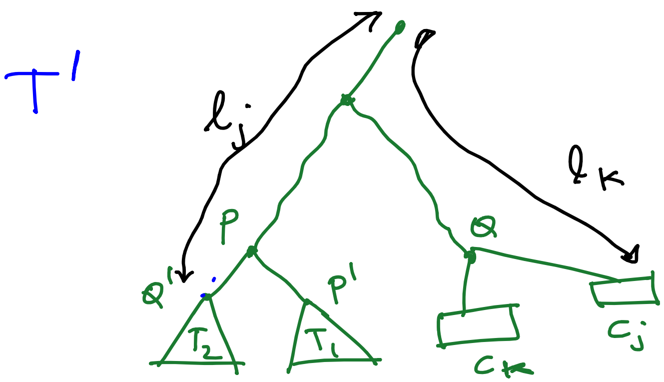



$l_j$ : length of the path from the root to $c_j$ (length of the code word for $c_j$)

$l_k$ : . . . .                . . .

$PP'$ and $QQ'$ are single edges.

Let $l_k \geq l_j$ (the case $l_j \geq l_k$ is symmetrical)




switched the positions of $c_j$ and $T_2$.

Let us assume the $wt(T_2) = \sum_{h=1}^{r} f_{i_h} \cdot d_{T_2}(c_{i_h})$

$\{c_{i_1}, \dots, c_{i_r}\}$ are the leaf nodes in $T_2$

$$wt(T') - wt(T)$$

$$= \sum_{h=1}^{r} f_{i_h}(l_j + d_{T_2}(c_{i_h})) + f_j l_k - f_j l_j - \sum_{h=1}^{r} f_{i_h}(l_k + d_{T_2}(c_{i_h}))$$

$$= l_j \underbrace{\sum_{h=1}^{r} f_{i_h}}_{d} + \cancel{\sum_{h=1}^{r} f_{i_h} d_{T_2}(c_{i_h})} + f_j l_k - f_j l_j - l_k \underbrace{\sum_{h=1}^{r} f_{i_h}}_{d} - \cancel{\sum_{h=1}^{r} f_{i_h} d_{T_2}(c_{i_h})}$$

$$= l_j d + f_j l_k - f_j l_j - l_k d$$

$$= (l_j - l_k)d - (l_j - l_k) f_j$$

$$= \underbrace{(l_j - l_k)}_{\leq 0} \underbrace{(d - f_j)}_{\geq 0}$$

$$\Rightarrow \quad \leq 0$$

All the leaves in set $\{c_{i_1}, \ldots, c_{i_r}\}$ have frequency $\geq f_j, f_k$

$\Rightarrow d - f_j \geq 0$

$\Rightarrow wt(T') \leq wt(T)$ ($\because T$ is optimal)

$\Rightarrow wt(T') = wt(T)$ $\square$

## Step 3

**Claim:** There is an optimal solution to C which arises by combining greedy choice (for c) and optimal solution to the subproblem $c'$.

**Proof:** By previous claim, there is an optimal solution S to C corresponding to greedy choice for c and solution $s'$ to subproblem $c'$.

$$\underbrace{wt_c(S)}_{\text{optimal}} = \overbrace{wt_{c'}(S')}^{\text{Also optimal}} + (f_j + f_k)$$

where $f_j$, $f_k$ are two least frequencies in C.

If $s'$ was not optimal then there is $s''$ (a solution of $c'$) s.t. $wt_{c'}(s'') < wt_{c'}(s')$

$\Longrightarrow$ there is a solution $R$ to $C$ which arises by combining choice $(f_j, f_k)$ with $S''$.

$$wt_C(R) = wt_{C'}(S'') + (f_j + f_k)$$

$$< wt_{C'}(S') + (f_j + f_k) = wt_C(S)$$

A contradiction, because $S$ is optimal solution to $C$.
(min weight)

$\Longrightarrow$ $S'$ is optimal solution of $C'$.

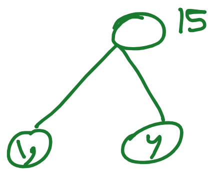
$\square$

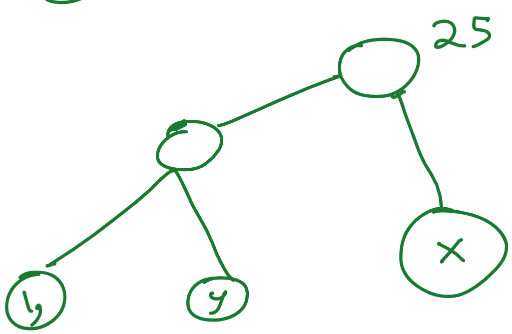
## Example

| a, | b, | x, | y, | r, | ƀ, | \\, | ← characters |
|---|---|---|---|---|---|---|---|
| 30 | 25 | 10 | 5 | 20 | 20 | 10 | ← frequencies |

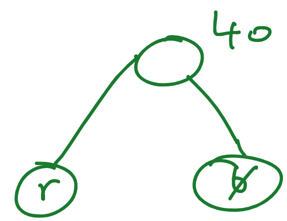
## Compute optimal code

| a | b | x | r | ƀ | |
|---|---|---|---|---|---|
| 30, | 25 | 10 | 20 | 20 | 15 |

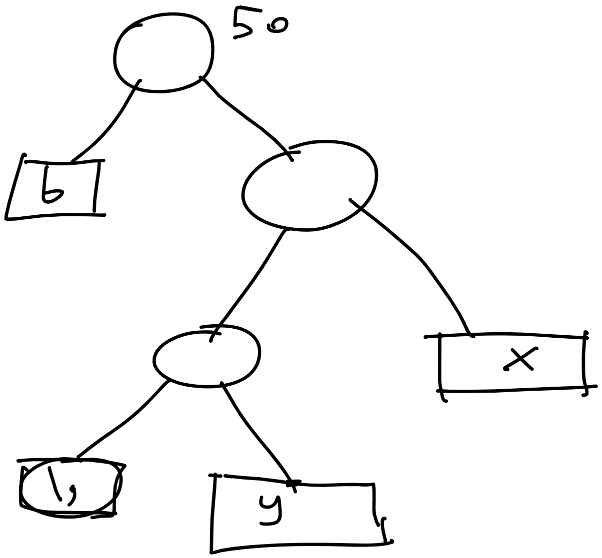




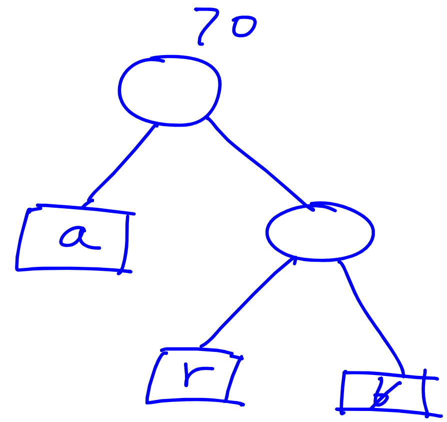
| a | b | r | ƀ | |
|---|---|---|---|---|
| 30 | 25 | 20 | 20 | 25 |



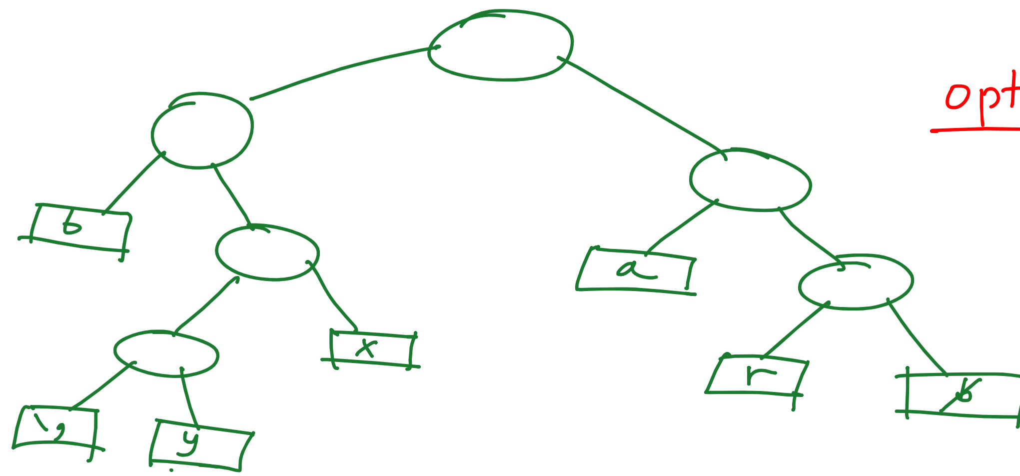
| a | b | | |
|---|---|---|---|
| 30 | 25 | 40 | 25 |




a
30 40 50




50 70




optimal tree

b → 00 - - -.

a → 10

\, → 0100

y → 0101

x → 011

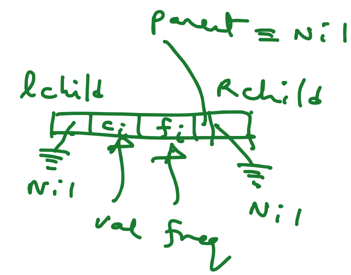
## Pseudo code

```
Huffman(C)
  n = |C|
  [Make priority queue Q of
  element of C with freq being the key
  for i = 1 to n-1 do
      x = min(Q)      // returns the min element
      delete(Q)       // deletes the min element from Q
      y = min(Q)
      delete(Q)
      z = new node for tree
      z.lchild = x
      z.rchild = y
      z.freq = x.freq + y.freq
      z.parent = nil
      insert(Q, z)
  // end for
  return min(Q)
```

- At any stage of the algorithm we have a set of trees.
- Each tree has an associated freq.
- Each time we need to pick two trees with min weight

Keep these trees in a priority queue.

Initially




Time complexity $O(n \log n)$