Huffman Codes

- Let $A_1,\ldots,A_n$ be a set of items.
- Let $P_1,\ldots,P_n$ be their probabilities.
- We want to find a set of lengths $L_1,\ldots,L_n$

that will produce a minimal $\sum_{i=1}^{n} L_i P_i$

Huffman's algorithm

If $(n==2)$ then $\{0,1\}$

Else combine 2 smallest probabilities P_n, P_{n-1}

solve for $P_1, P_2, \ldots, P_{n-2}, P_{n-1}+P_n$

if $P_{n-1}+P_n$ is represented by α

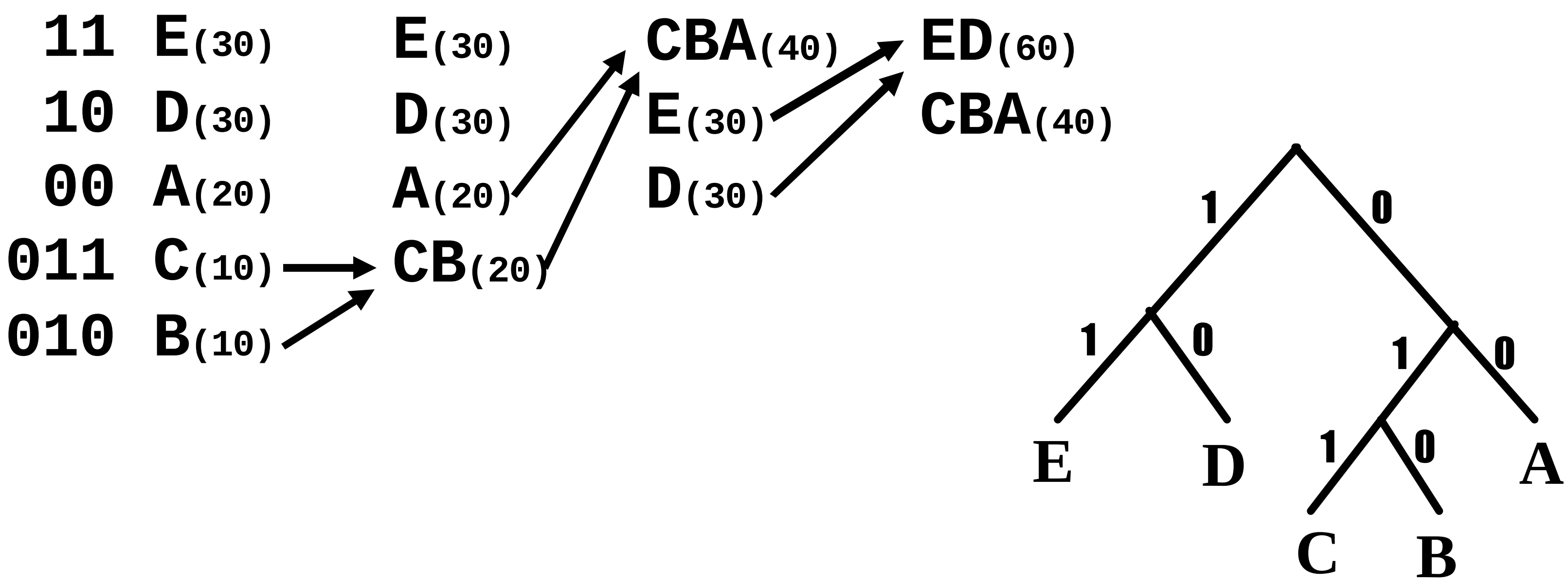
then

P_{n-1} will be represented by $\alpha 0$

P_n will be represented by $\alpha 1$

Example of Huffman coding

- In a given language:
  - **A - 20%, B - 10%, C - 10%, D - 30%, E - 30%**

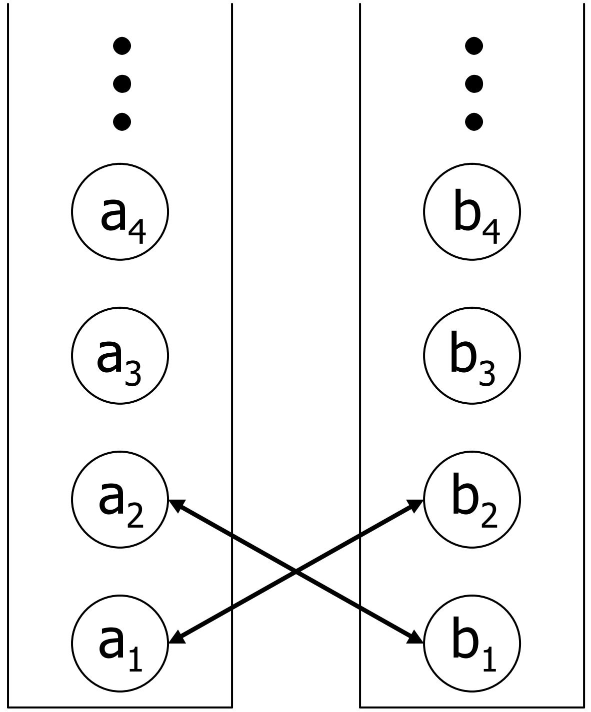


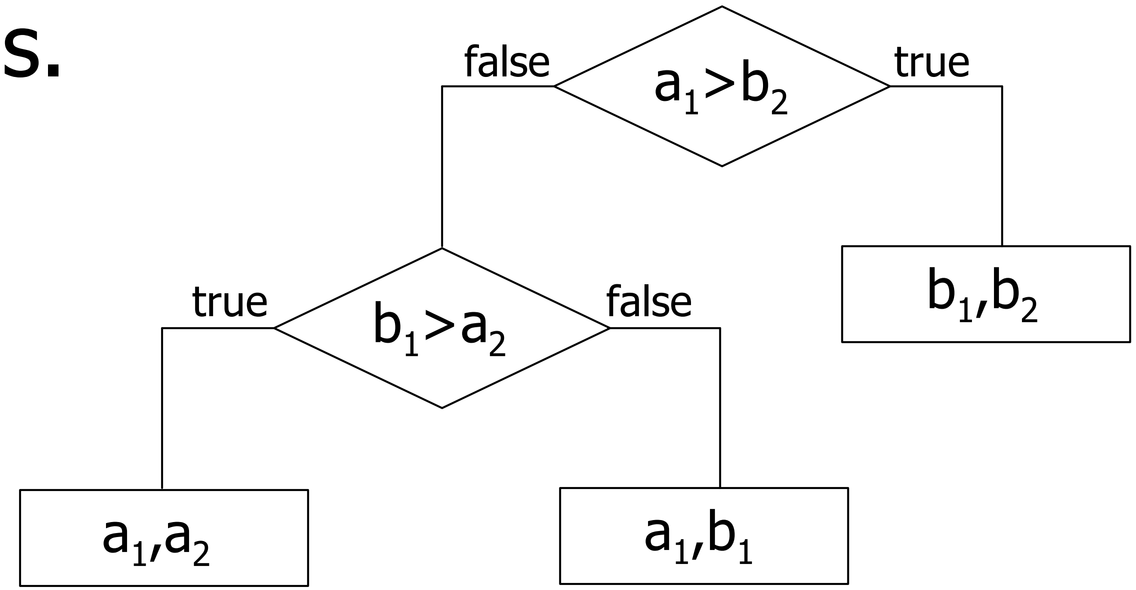
Implementation

- Sorting is O(n log n).
- Finding where to insert a new item is O(n).
  - There are n-2 items to find, so the total order is $O(n^2)$.
- Total order is $O(n\log n)+O(n^2)$ i.e. $O(n^2)$.

Improved implementation

- Two queues.





Improved implementation (cont.)

Sort all items into queue "a".

Queue "b" is empty.

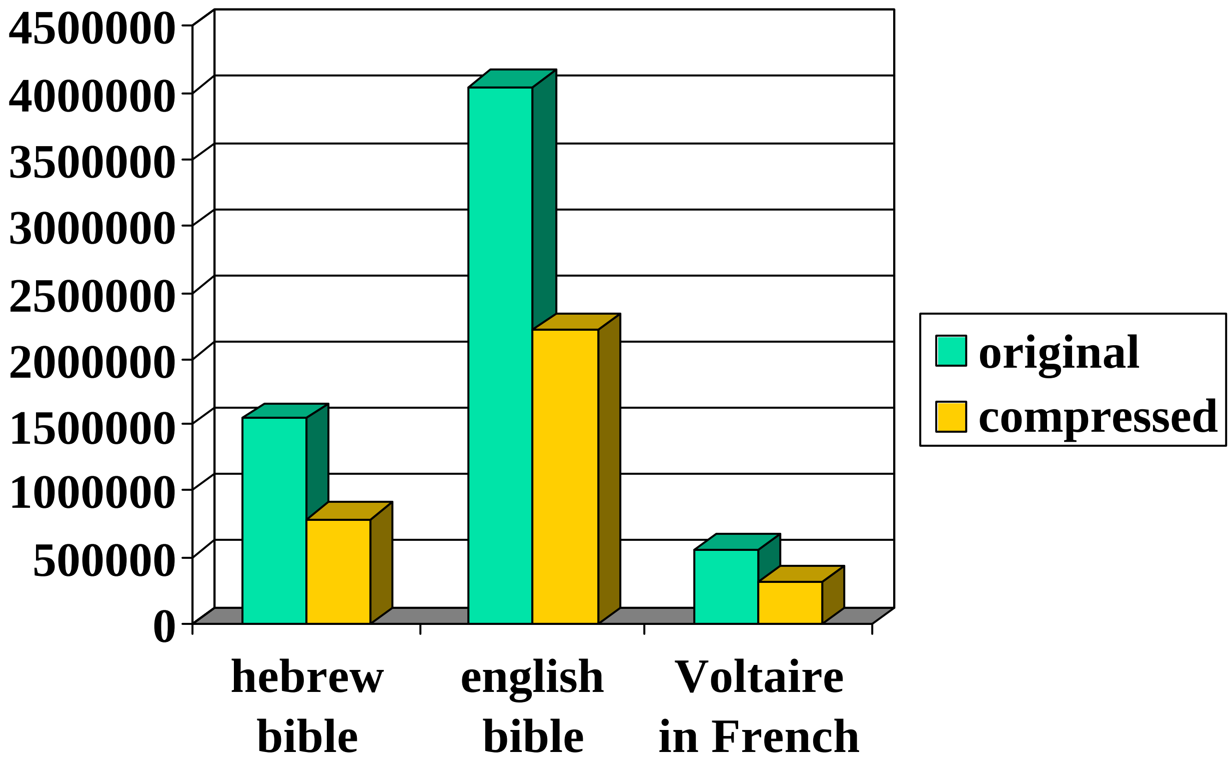
While there are values in the queues,

    Put sum of 2 low values in queue "b".

- The order of the numbers created by the additions are in non-decreasing order.
- The algorithm is O(nlogn)+O(n) i.e. O(nlogn)

Results of Huffman

- Examples for text files:



Huffman is optimal

- An optimal tree is a tree for which

$$\sum_{i=1}^{n} L_i P_i$$

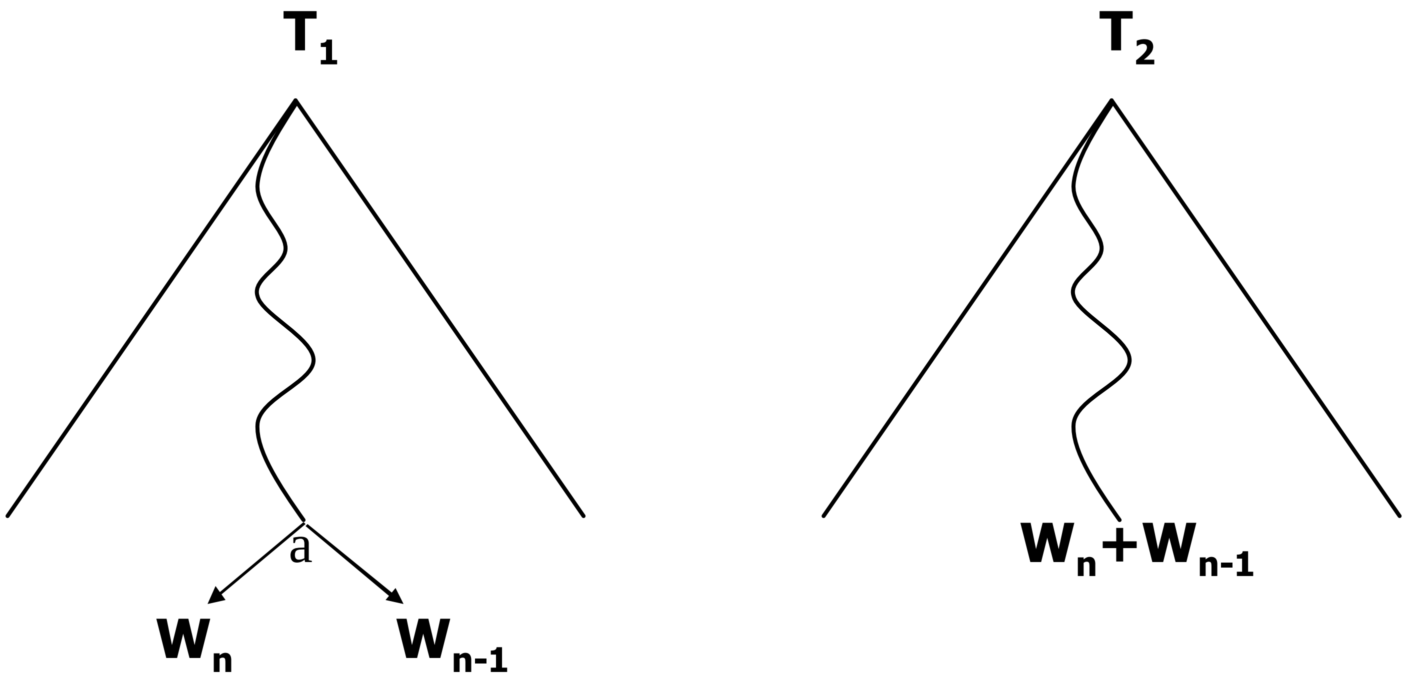
is minimal

- Lemmas: In an Optimal tree:
  - The tree is full so at least 2 nodes on lowest level.
  - 2 nodes with lowest weights are on lowest level.
  - 2 lowest weights can be on brother nodes.

Theorem

- Let T_1 be an optimal tree with weights $W_1,\ldots,W_n$.
- The two lowest weights W_n,W_{n-1} are on lowest level and they are brothers.
- Let α be the father of W_n,W_{n-1}.
- Let T_2 be a same tree as $T_{1,,}$, without Wn,Wn-1 but with α having weight W_n+W_{n-1}
- **Theorem**: T_2 is an optimal tree.

Theorem (cont.)



Theorem - proof

- Let us denote $M_1 = \sum_{i=1}^{n} L_i W_i$.

- $M_1 = W_1 L_1 + \ldots + W_{n-1} L_{n-1} + W_n L_{n-1}$

  Brothers have same level

- $M_2 = M_1 - (W_{n-1} L_{n-1} + W_n L_{n-1}) + (W_{n-1} + W_n)(L_{n-1} - 1)$

  $= M_1 - W_{n-1} - W_n$

  Weight of α

Theorem - proof (cont.)

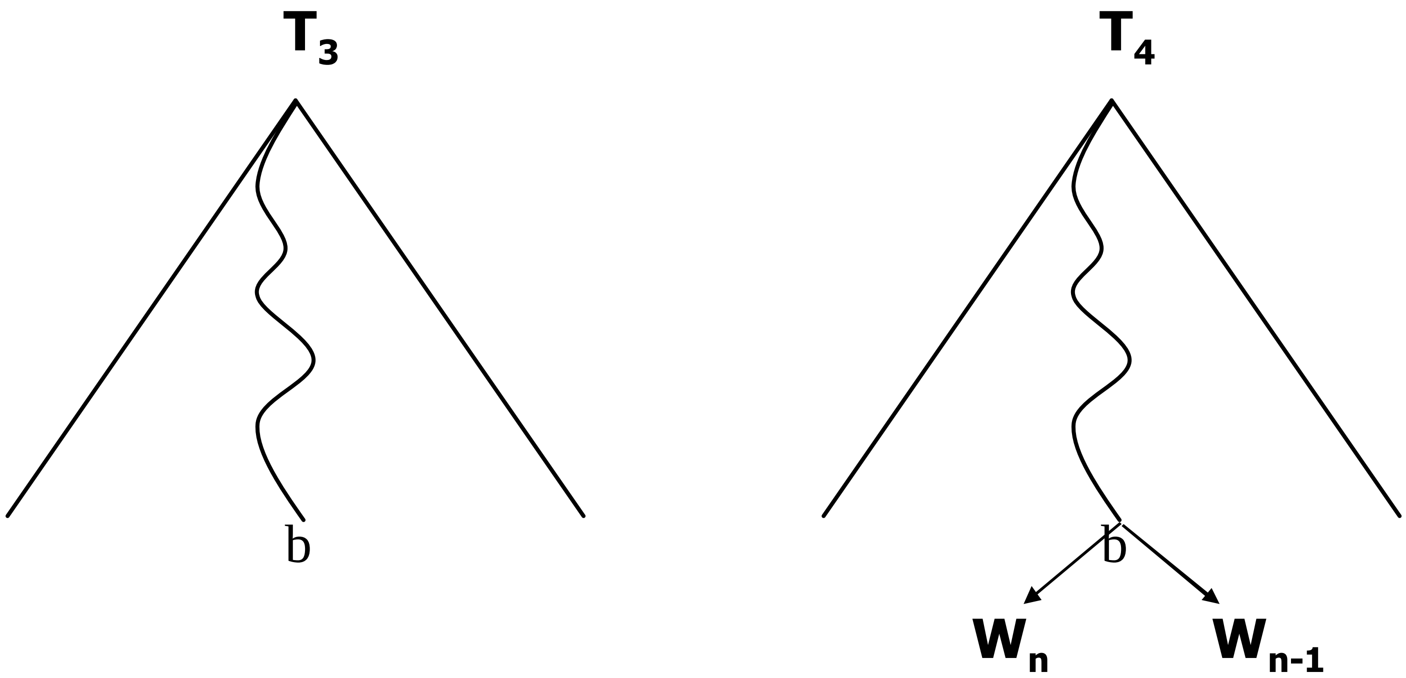
- Suppose T_2 is not Optimal.

$\Rightarrow$There is a tree $T_3 \neq T_2$ which is optimal.

$\Rightarrow M_3 < M_2$.

- There is a leaf β in T_3 which has weight $W_{n-1}+W_n$.
- Let T_4 be a tree with same nodes as T_3 but instead of β it has two leaves - W_{n-1}, W_n

Theorem - proof (cont.)



Theorem - proof (end)

- T_4 is a tree for the weights $W_1,\ldots,W_n$.
- $M_4=M_3+W_{n-1}+W_n<M_2+W_{n-1}+W_n=M_1$

$\Rightarrow M_4<M1$

- But M_1 is an optimal tree. contradiction!

$\therefore T_2$ is an optimal tree.

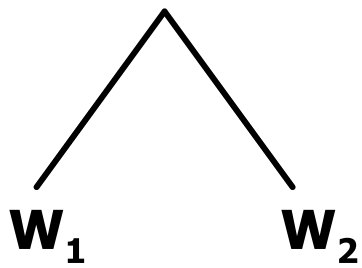
Theorem

- Let T be an optimal tree for any n leaves.

$\Rightarrow$T is equivalent to a Huffman tree with same leaves. i.e. they both have same $\sum_{i=1}^{n} L_i W_i$.

Theorem - Proof

- Induction on number of leaves.
- for n=2 there is just one option so the trees are equivalent.



Theorem proof (end)

- Let us assume for n-1 and prove for n.
  - Let T_1 be an optimal tree.
  - According to previous theorem T_2 is optimal too.
  - According to induction's assumption T_2 which has n-1 leaves, is equivalent to a Huffman tree.
  - T1 was built according to Huffman's algorithm so T1 is equivalent to a Huffman Tree too.

Optimal trees

- Huffman trees are optimal as proved but there are other trees which are optimal but not Huffman.
- Example:
  - A 20%, B 10%, C 10%, D 30%, E 30%