perché la dimensione del file compresso è

$$B(T) \cdot \frac{|F|}{100}$$

$|F|$ → n° caratteri file non compresso

$\Rightarrow$ $B(T)$, a meno di una costante (100), è ~~il fattore di~~ l'inverso del fattore di compressione: minore è $B(T)$ e migliore è il fattore di compressione.

(Osservazione: stiamo assumendo di dover associare una codeword a ogni singolo carattere, il che non è sempre la cosa migliore da fare:

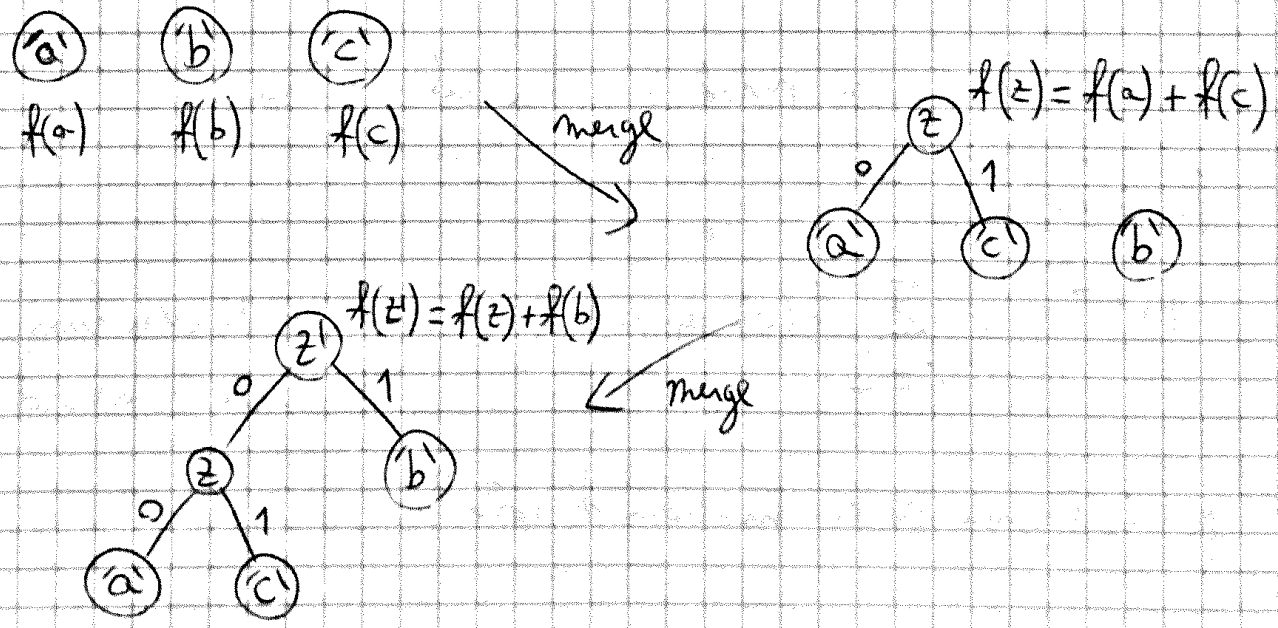
abbabbabbabb....

Cioè, vedremo il codice libero da prefissi ottimo tra tutti i codici che associano una codeword a ogni singolo carattere)

Osservazione: un albero ottimo è sempre pieno (cioè i nodi interni hanno 2 figli)

$\Rightarrow$ spazio sottoproblemi: alberi pieni a n foglie

Goal: trovare un albero ottimo T, cioè che minimizza $B(T)$



sottoproblemi = alberi su meno caratteri

scelta greedy = merge tra nodi che hanno frequenza minore

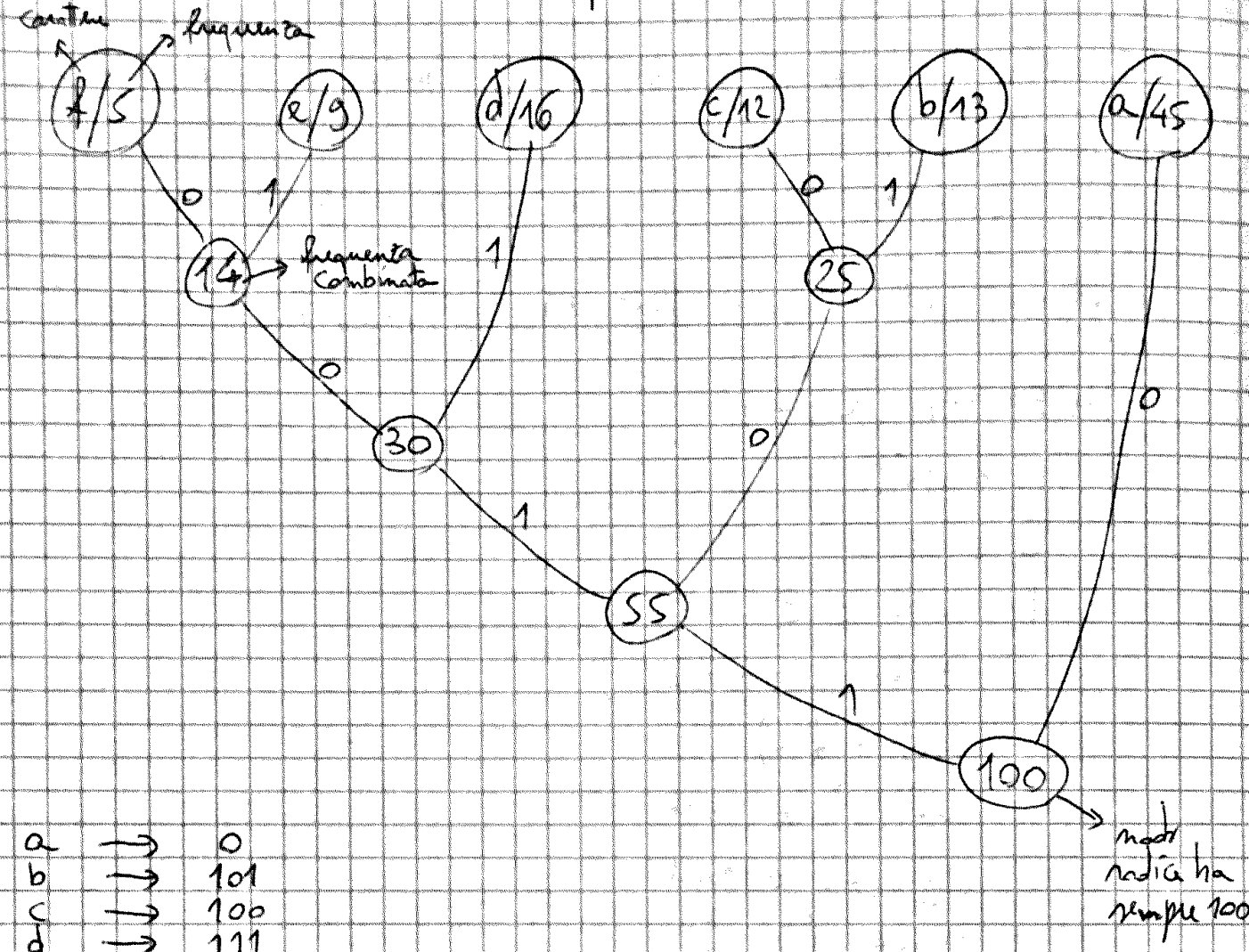
Q = coda di priorità di nodi con chiave $f(z)$

attributi del nodo z: left[z], right[z], f(z)

```
HUFFMAN (C, f)
    n ← |C|
    Q ← ∅
    for each c ∈ C do
        z ← NEW_NODE()
        f[z] ← f(c)
        left[z] ← null
        right[z] ← null
        INSERT (Q, z)
    for i ← 1 to n-1 do
        x ← EXTRACT_MIN (Q)
        y ← EXTRACT_MIN (Q)
        z ← NEW_NODE()
        left[z] ← x
        right[z] ← y
        f[z] ← f[x] + f[y]
        INSERT (Q, z)
    return EXTRACT_MIN (Q)
```

Complessità: $\sum_{i=1}^{n-1} \Theta(\log(n-i)) = \Theta(n \log n)$

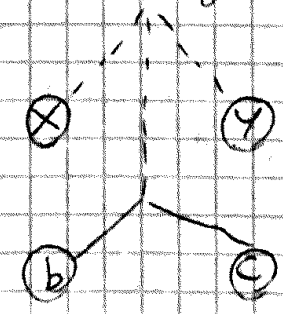
Funzionamento sul nostro esempio:



a → 0
b → 101
c → 100
d → 111
e → 1101
f → 1100

<u>Proprietà di scelta greedy</u>: sia C un alfabeto e siano X e Y i caratteri in C di frequenza minore. Allora esiste un codice prefisso ottimo T in cui X e Y sono foglie attaccate allo stesso padre.

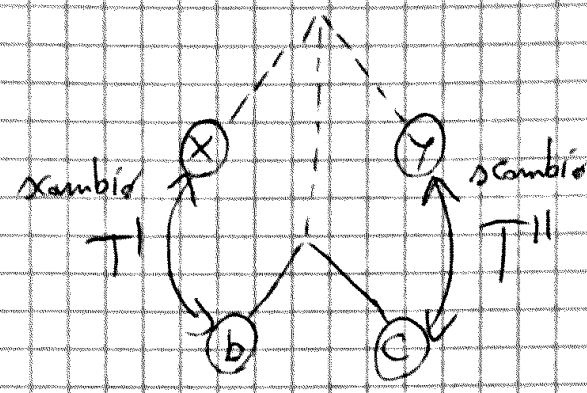
Dimostrazione: sia T^* una soluzione ottima
siano b e c le foglie di profondità max in T^*



1) $d_{T^*}(b) = d_{T^*}(c) \geq d_{T^*}(x),\ d_{T^*}(y)$

2) $f(x) \leq f(b)$

$f(y) \leq f(c)$



$$T^* \longrightarrow T' \longrightarrow T''$$

devo mostrare che $B(T^*) \geq B(T') \geq B(T'')$

$T^* \longrightarrow T'$: $B(T^*) - B(T') = \sum_{c \in C} f(c)\, d_{T^*}(c) +$

$- \sum_{c \in C} f(c)\, d_{T'}(c) = f(b)\, d_{T^*}(b) +$

e poi lo stesso per $T' \longrightarrow T''$

$+ f(x)\, d_{T^*}(x) - f(b)\, d_{T'}(b) - f(x)\, d_{T'}(x) =$

ma $d_{T'}(b) = d_{T^*}(x)$, $d_{T'}(x) = d_{T^*}(b)$

$\Rightarrow = f(b)\, d_{T^*}(b) + f(x)\, d_{T^*}(x) - f(b)\, d_{T^*}(x)$

$- f(x)\, d_{T^*}(b) = f(b)\left(d_{T^*}(b) - d_{T^*}(x)\right)$

$- f(x)\left(d_{T^*}(b) - d_{T^*}(x)\right) =$

$$= \underbrace{\left(d_{T^*}(b) - d_{T^*}(x)\right)}_{\geq 0} \underbrace{\left(f(b) - f(x)\right)}_{\geq 0}$$

$\Rightarrow B(T^*) = B(T')$