Activity Selection (Interval scheduling)

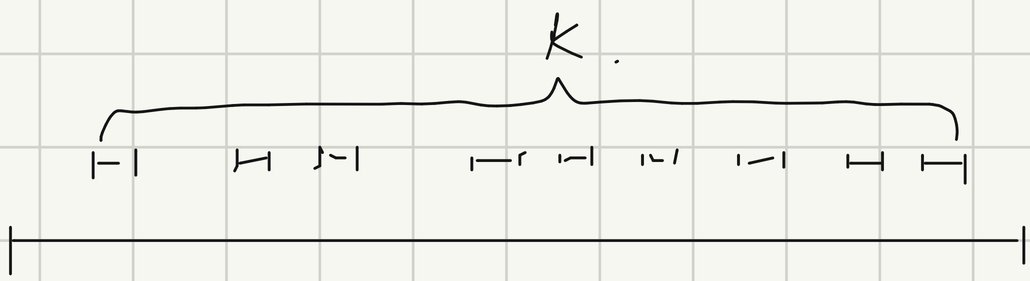
Input: a set of activities $(s_1, f_1), \cdots, (s_n, f_n)$

Output: a maximum set of mutually compatible activities

compatible → non-overlapping

rule 1: smallest start time

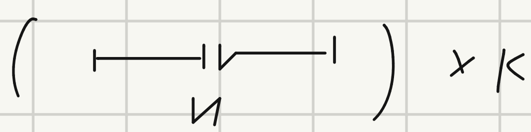
(what if an activity start earlist but last very long?)



OPT: K (optimal)

Greedy = 1

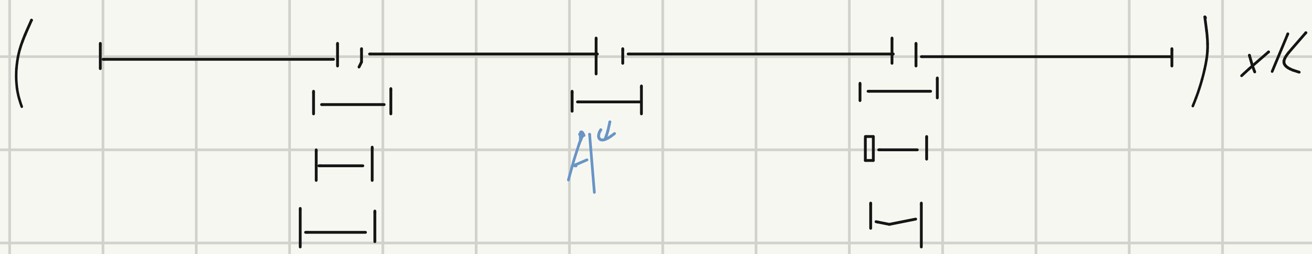
rule 2: shortest interval



OPT = 2K

Greedy = K

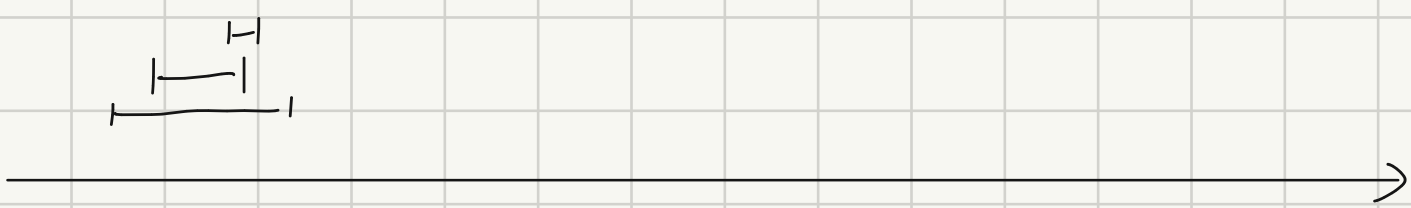
rule 3: fewest conflict



OPT = 4K

Greedy = 3K

Greedy → first select "A", which has the least 2 conflicts



rule 4: earliest finish time

1. let R be the set of activities.
2. let $A = \phi$
3. sort R by finish time
4. for activity $i \in R$.
5. if i is compatible with A
6. add i to A.
7. return A

1–2: $O(1)$

3: $O(n\log n)$

4: $\times n$

5: $O(1)$ (i beginning ≥ A's last finish)

$\Rightarrow O(n\log n)$

Theorem:

A is optimal.

Proof:

Suppose A is not optimal. $|OPT| > |A|$.

A: $i_1, i_2, \ldots, i_k$.

OPT: $j_1, j_2, \ldots, j_k, j_{k+1} \ldots j_m$

"exchange argument"

$i_1, i_2, i_3, \ldots, i_t$

$\parallel \quad \parallel \quad \parallel \quad \neq$

$j_1, j_2, j_3, \ldots, j_t$

j_t substituted by i_t $\Rightarrow$ safe

greedy $\Rightarrow f_{i_t} \le f_{j_t}$ $\quad f$: finish time.

then OPT': $i_1, i_2, \ldots, i_k, j_{k+1}, \ldots, j_m$

So why Greedy doesn't include $j_{k+1}, \ldots, j_m$?

$\Rightarrow$ Contradiction

$\Rightarrow$ So A is optimal

Data Compression

$\Sigma = \{A, B, C, D\}$

A	00
B	01
C	10
D	11

$\Rightarrow$

A	0
B	01
C	10
D	1

avg(length) = 1.5.

problem: when decoding, the digit width confused

Prefix:

for a string $S = a_1 \cdots a_n$

for $i = 0, \cdots, n$

$a_1 a_2 \cdots a_i$ is a prefix of S

A:	0
B:	10
C	110
D	111

ABAD

0 10 0 111

$\overline{A}$ $\overline{B}$ $\overline{A}$ $\overline{D}$.

One is not another one's prefix, then eliminate confusion.

A prefix (free) code for an alphabet Σ, r is a function

$r: \Sigma \to \underbrace{\{0,1\}}_{}^{*}$ such that for any $a, b \in \Sigma$, $r(a)$ is not prefix of $r(b)$.

↳ string consist of 0, 1.

$f(A) = 0.6$

$f(B) = 0.2$ $\Rightarrow$ $1.6.$ $\quad 0.6\times1 + 0.2\times2 + 0.1\times3 + 0.1\times3 = 1.6$

$f(C) = 0.1$

$f(D) = 0.1$

Input: An alphabet Σ with frequence f_a for each $a \in \Sigma$

Assume $|\Sigma| \geq 2$.

Output: a prefix code r for Σ that minimizes

$$\sum_{a\in\Sigma} |r(a)| f(a)$$

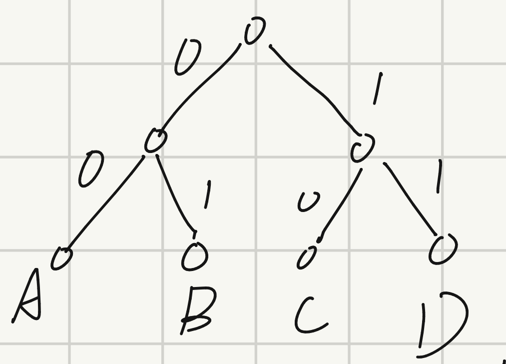
code $\Leftrightarrow$ binary tree

A 00

B 01

C 10

D 11

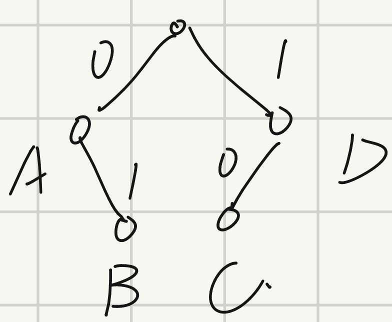


A: 0

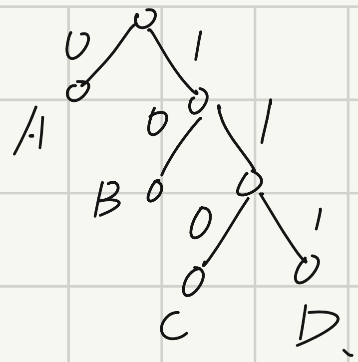
B: 01

C: 10

D 1



A	0
B	10
C	110
D	111



$\Rightarrow$ can be used as decoder

0100111

prefix code r $\Leftrightarrow$ a binary tree where only leaves are $\underbrace{\text{labelled by distinct}}_{+}$ symbol

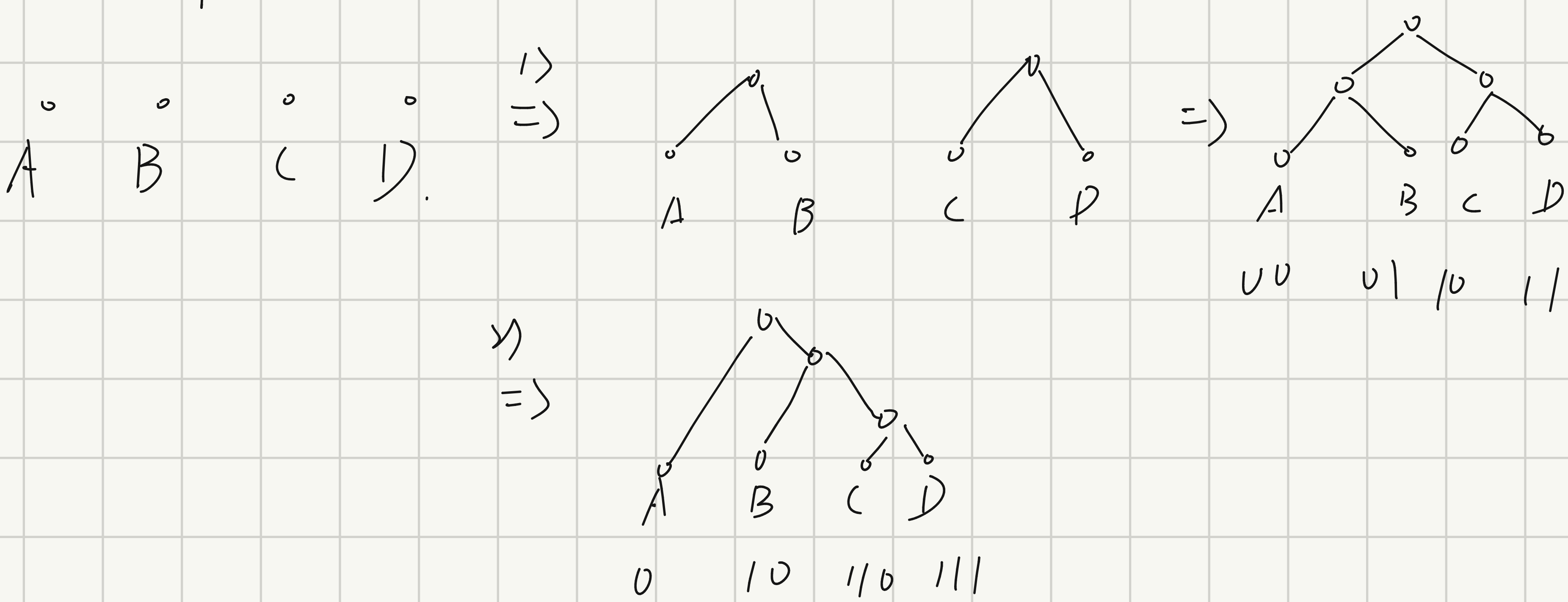
Σ- tree.

$$\sum_{a \in \Sigma} |r(a)| f(a) \quad \Leftrightarrow \quad \sum_{a \in \Sigma} \text{depth}(a) f(a)$$

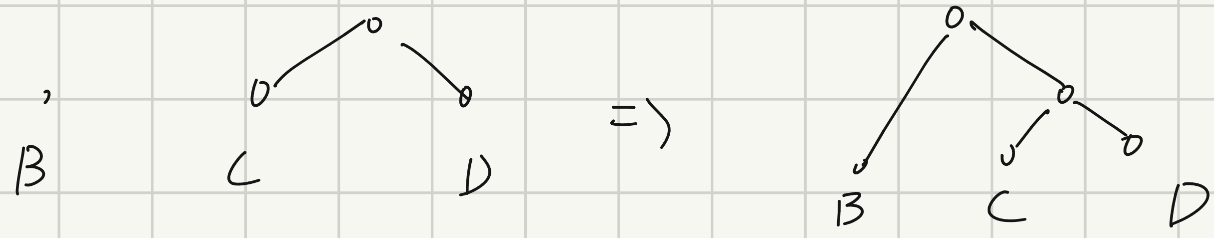
Input : Σ and $\{f_a\}_{a \in \Sigma}$

Output : a Σ-tree that minimizes $\sum_{a \in \Sigma} \text{depth}(a) f_a = c(T)$

bottom-up construction.



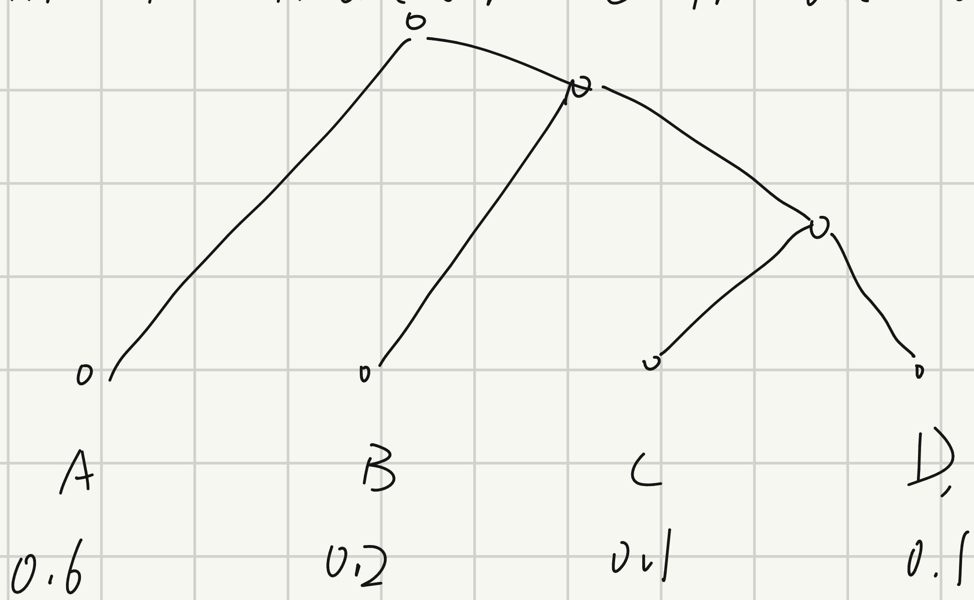
$\circ = f_c + f_d.$



$\triangle = f_B + f_c + f_d.$

Huffman's criterion:

minimum increasement in the average leaf depth.



Huffman's Algorithm

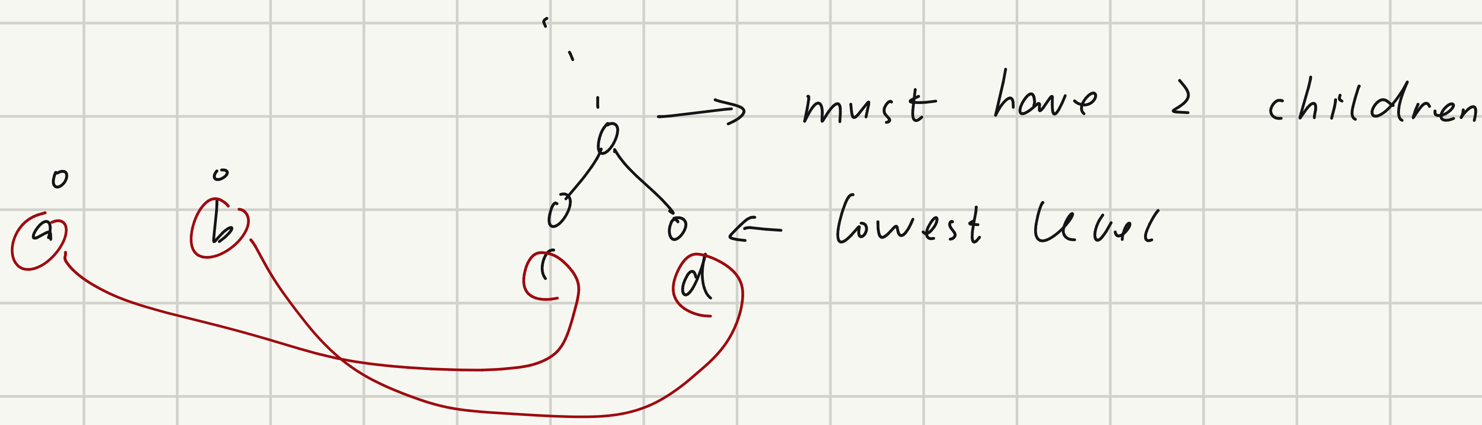
```
for each a ∈ Σ
    create a tree Ta of a single node a.
    f(Ta) = fa.
let F be the set of all trees
while |F| >= 2
    let T1 and T2 be the two trees with minimum freq.
    F := F - {T1, T2}
    T3 := merge {T1, T2}
    f(T3) := f(T1) + f(T2)
    add T3 to F
return the tree remaining in F
```

Lemma:

let a, b be the symbol with min freq.

There is an optimal tree in which a and b are siblings

Proof: OPT



$f_a \le f_c$.

$f_b \le f_d$.

Theorem:

Huffman's algorithm gives an optimal Σ-tree.

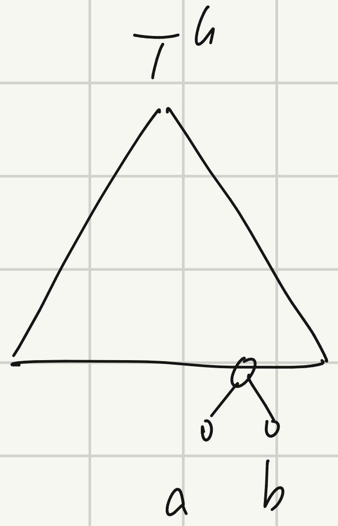
Proof:

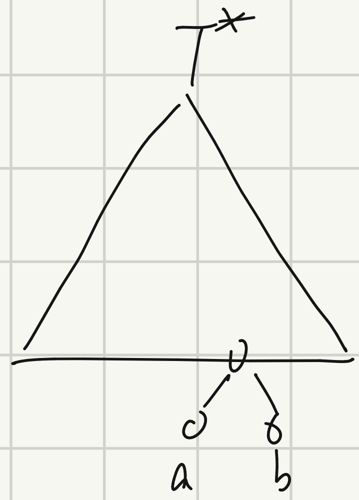
By Induction on $|\Sigma|$

base case: $|\Sigma| = 2$. optimal

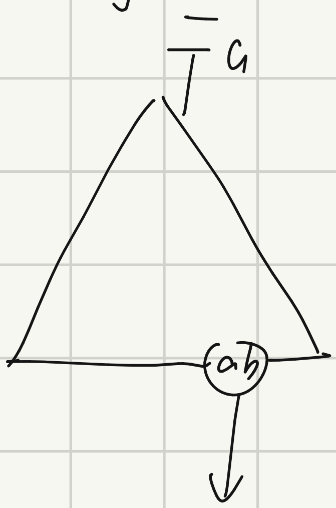
Inductive hypothesis: assume when $|\Sigma| = k$, optimal

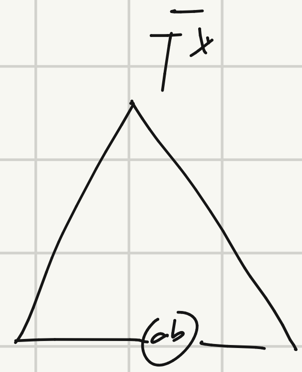
Inductive step: when $|\Sigma| = k+1$





goal : $c(T^G) \le c(T^*)$





$f(ab) = f_a + f_b$

$c(\bar{T}^G) = c(T^G) - f_a - f_b.$ $\quad c(\bar{T}^*) = c(T^*) - f_b - f_a$

prove $\quad c(\bar{T}^G) \le c(\bar{T}^*)$

$\bar{T}_G$ is Σ-tree for $\Sigma' = \Sigma - \{a, b\} + \{\overset{\text{metasymbol}}{ab}\}$

$|\Sigma'| = |\Sigma| - 1 = k$

Then use inductive hypothesis,

eg: