Κωδικοποίηση Huffman:

Huffman(f):

Είσοδος: ακολουθία $f[1,\dots,n]$ συχνοτήτων

Έξοδος: Δέντρο κωδικοποίησης με n-φύλλα

$H = \emptyset$ (ουρά προτεραιότητας με κλειδιά τις συχνότητες)

πεδίο κλειδί

for $i = 1$ to n

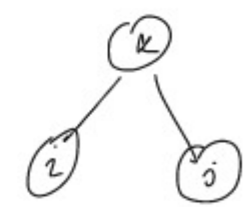
  insert $(H, i, f[i])$

$O(n \log n)$

for $k = n+1$ to $2n-1$

  i = deletemin (H), j = deletemin (H)

  φτιάξε κόμβο k με παιδιά i, j ⇝



  $f[k] = f[i] + f[j]$

  insert $(H, k, f[k])$

Π.χ.

A → 1

B → 2

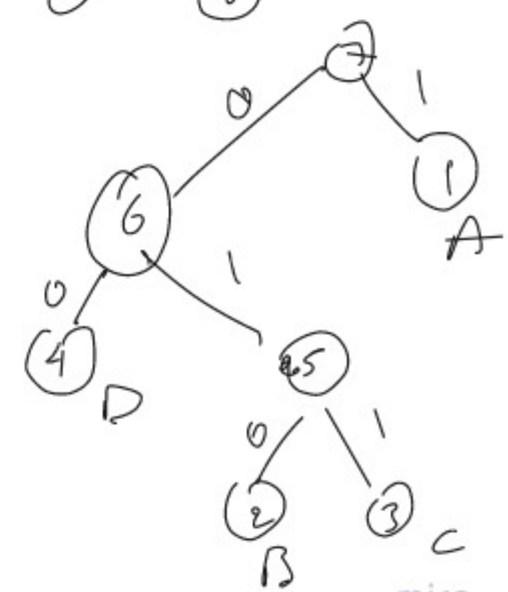
C → 3

D → 4

$H = \begin{array}{cccc} 54 & 2 & 15 & 29 \\ [1, & \cancel{2}, & \cancel{3}, & 4] \end{array}$

$H = \begin{array}{ccc} 54 & 29 & 17 \\ [1, & \cancel{4}, & \cancel{5}] \end{array}$

$H = \begin{array}{cc} 54 & 46 \\ [\cancel{1}, & \cancel{6}] \end{array}$



44 100 відліків за сек.

$s_1, s_2, \ldots, s_T$

130 симв.

20 Hz

20 kHz

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

$s_1, s_2, \ldots, s_T$

ACAB

A → 00

B → 01

C → 10

D → 11

260 megabit

A	70 εκ	54 %
B	3 εκ	2 %
C	20 εκ	15 %
D	37 εκ	29 %

λάθος τρόπος !

A → 0

B → 01

C → 11

D → 001

ACAD $\rightsquigarrow$ $\underbrace{0}_{A}\ \underbrace{101}_{C}\ \underbrace{0}_{A}\ \underbrace{11}_{D}$

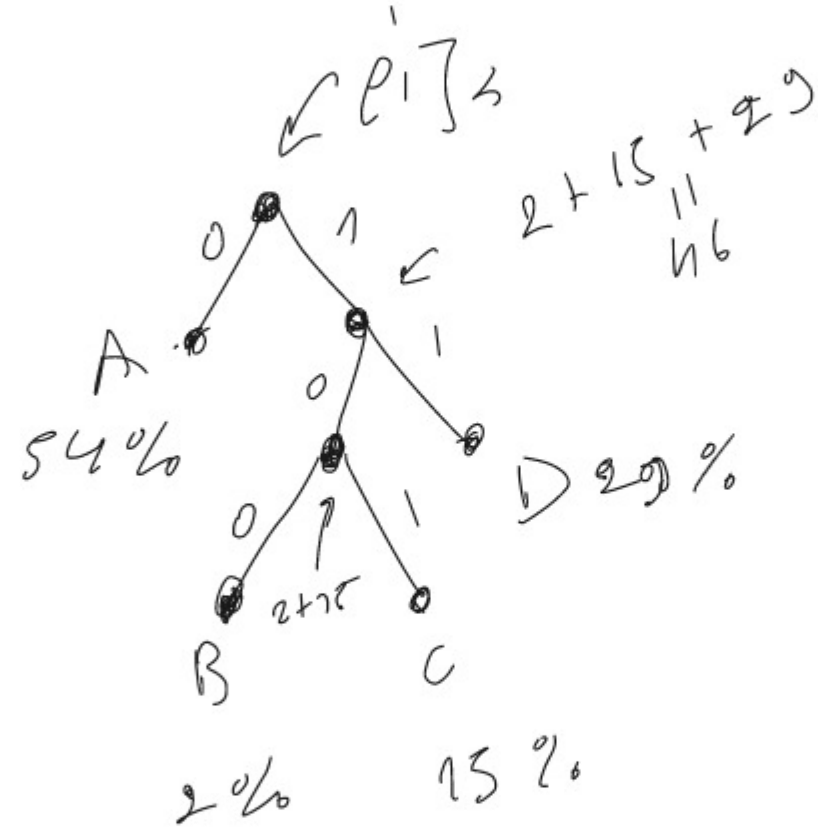
[illegible]:

A $\rightsquigarrow$ 0

B $\rightsquigarrow$ 100

C $\rightsquigarrow$ 101

D $\rightsquigarrow$ 11



κόστος δέντρου!

$$54 \cdot 1 + 2 \cdot 3 + 15 \cdot 3 + 29 \cdot 2 =$$

$$= \ldots$$

n-σύμβολα, f_i συχνότητα του συμβόλου i

$$\text{κόστος} = \sum_{i=1}^{n} f_i \cdot (\text{βάθος } i\text{-οστού συμβόλου})$$

Κόλπος! Ιδέα για σχηματισμό δέντρων: κορυφή-το δέντρο επιστρέφοντας τη ρίζα.

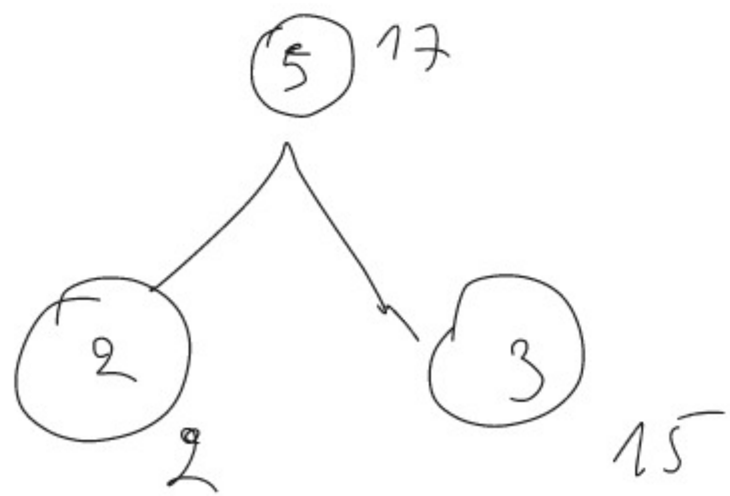
A → 1

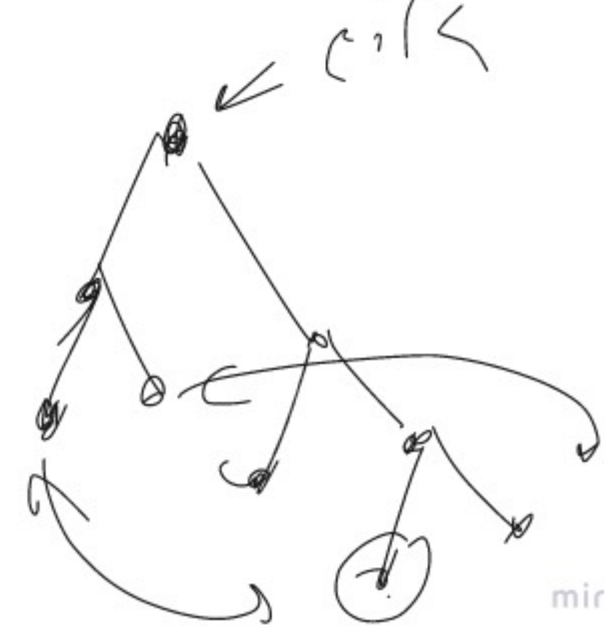
~~B → 2~~

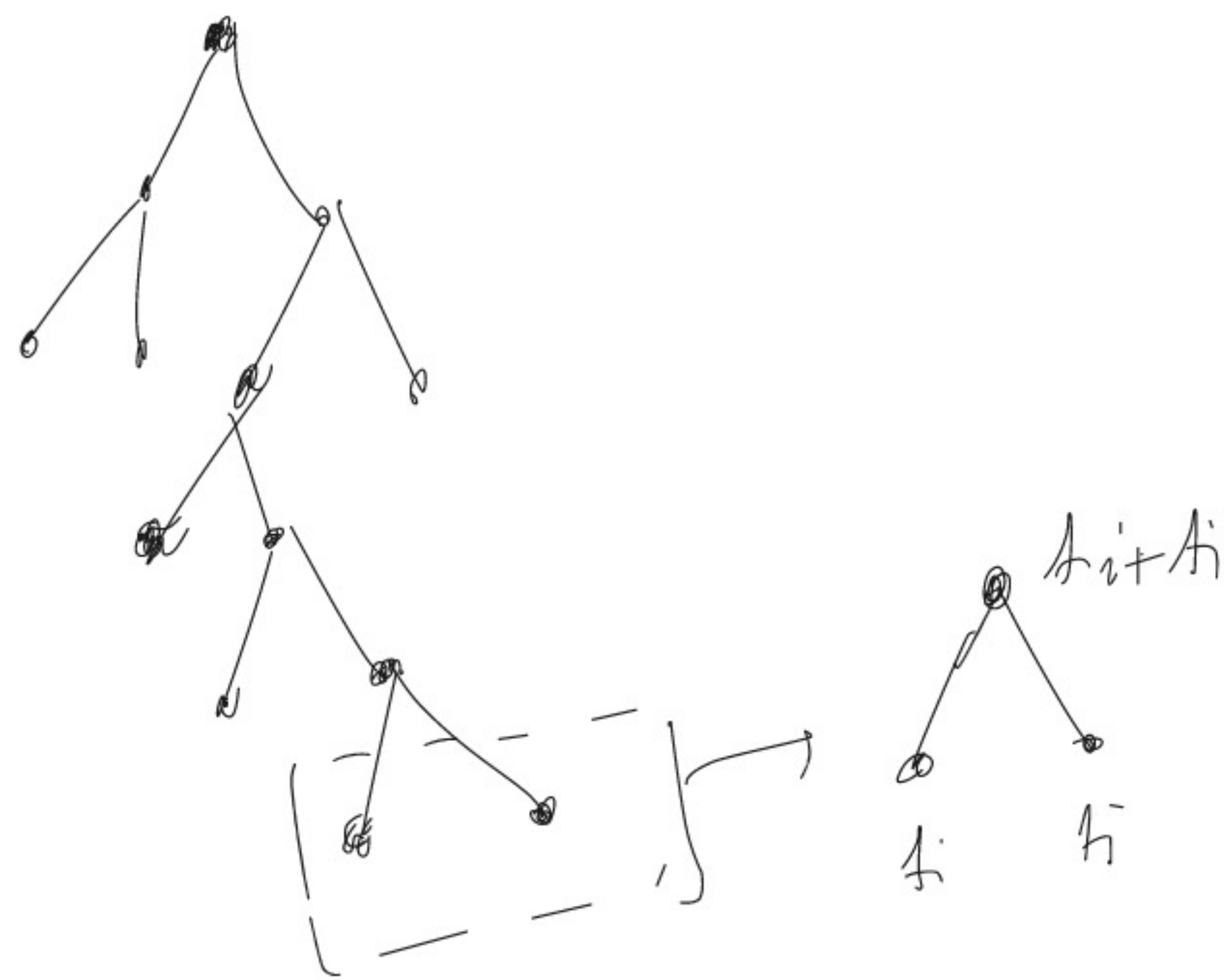
~~C → 3~~

C → 4

S






fi+fj
fi
fj

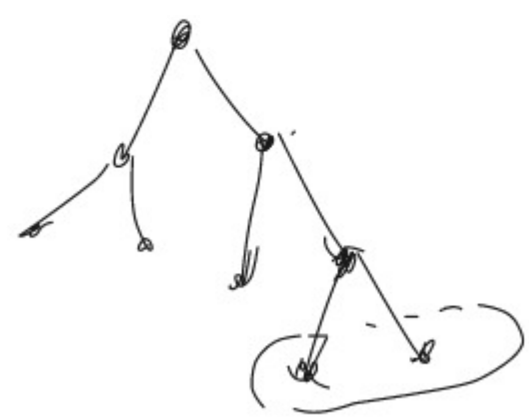
Ένα δέντρο Huffman με n-φύλλα έχει $2n-1$ κόμβους.

Επαγωγή στο n

<u>Βάση:</u> $n = 1$ ισχύει

<u>Ε.Υ.:</u> ισχύει με $n = k$

<u>Βήμα!</u> Δέντρο Huffman με $(k+1)$-φύλλα. $\to 2(k+1) - 1 = 2k+1$.



$\to 2k - 1 + 2 =$

$= 2k + 1$

Τύποι Horn;

Horn(S):

$c_1 \wedge \dots \wedge c_m$ — όροι

$x_1, \dots, x_n$

Είσοδος: Τύπος Horn $S=\{c_1,\dots,c_m\}$ (με n-μεταβλητές)

Έξοδος: Πίνακας V με τιμές T, F για τις n-μεταβλητές) να ικανοποιεί τον S ή μήνυμα σφάλματος αν δεν ικανοποιείται ο S

```
for i=1 to n
    V[i] = F
for every c ∈ S
    if c = x_i      (x_i μεταβλητή)
        V[i] = T ,  S = S \ {c}
```

$O(m)$

```
consistent = T
change = T
while consistent = T and change = T
    change = F
    for every c ∈ S
        if c = ¬x_{i1} ∨ ... ∨ ¬x_{im}  and V[i1] = ... = V[im] = T
            consistent = F
        if c = x_{i1} ∧ ... ∧ x_{im} → x_j  and V[i1] = ... = V[im] = T and V[j] = F
            V[j] = T
            change = T
            S = S \ {c}
```

$\leftarrow O(n^2)$

*

*

```
if consistent = F
    return 'Μη ικανοποιήσιμος'
else return V.
```

χρόνος $O(n^2)$.

προτασιακές μεταβλητές

$x_1, x_2, \ldots$ $\quad$ (x_1 = ο φόνος έγινε στην κουζίνα)

στοιχεία

$x_1, x_2, \ldots$

$\bar{x}_1, \bar{x}_2, \ldots$ $\quad$ ($\bar{x}_1$ = ο φόνος <u>δεν</u> έγινε στην κουζίνα)

$\wedge, \vee, \rightarrow, \neg \quad \rightsquigarrow \quad (x_1 \vee x_2) \wedge x_3 \rightarrow x_4$

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ T & F & T & F \end{array}$$

X	Y	$X \rightarrow Y$
T	T	T
T	F	F
F	T	T
F	F	T

Τύπος Horn:

όροι:
- x_i μεταβλητές
- $x_{i_1} \wedge \dots \wedge x_{i_m} \to x_j$
- $\bar{x}_{i_1} \vee \dots \vee \bar{x}_{i_m}$

ορ. Τύπος Horn [illegible] [illegible]

π.χ. $s = (x_4 \wedge x_2 \wedge x_3 \to x_1) \wedge (x_1 \wedge x_3 \to x_4) \wedge (x_1 \to x_2) \wedge$
$\wedge \cancel{(x_1)} \wedge (x_1 \wedge \cancel{x_2} \to x_4) \wedge (\bar{x}_4 \vee \bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_3)$

	x_1	x_2	x_3	x_4
$v =$	[F	F	F	F]
$v =$	[T	F	F	F]
$v =$	[T	T	F	F]

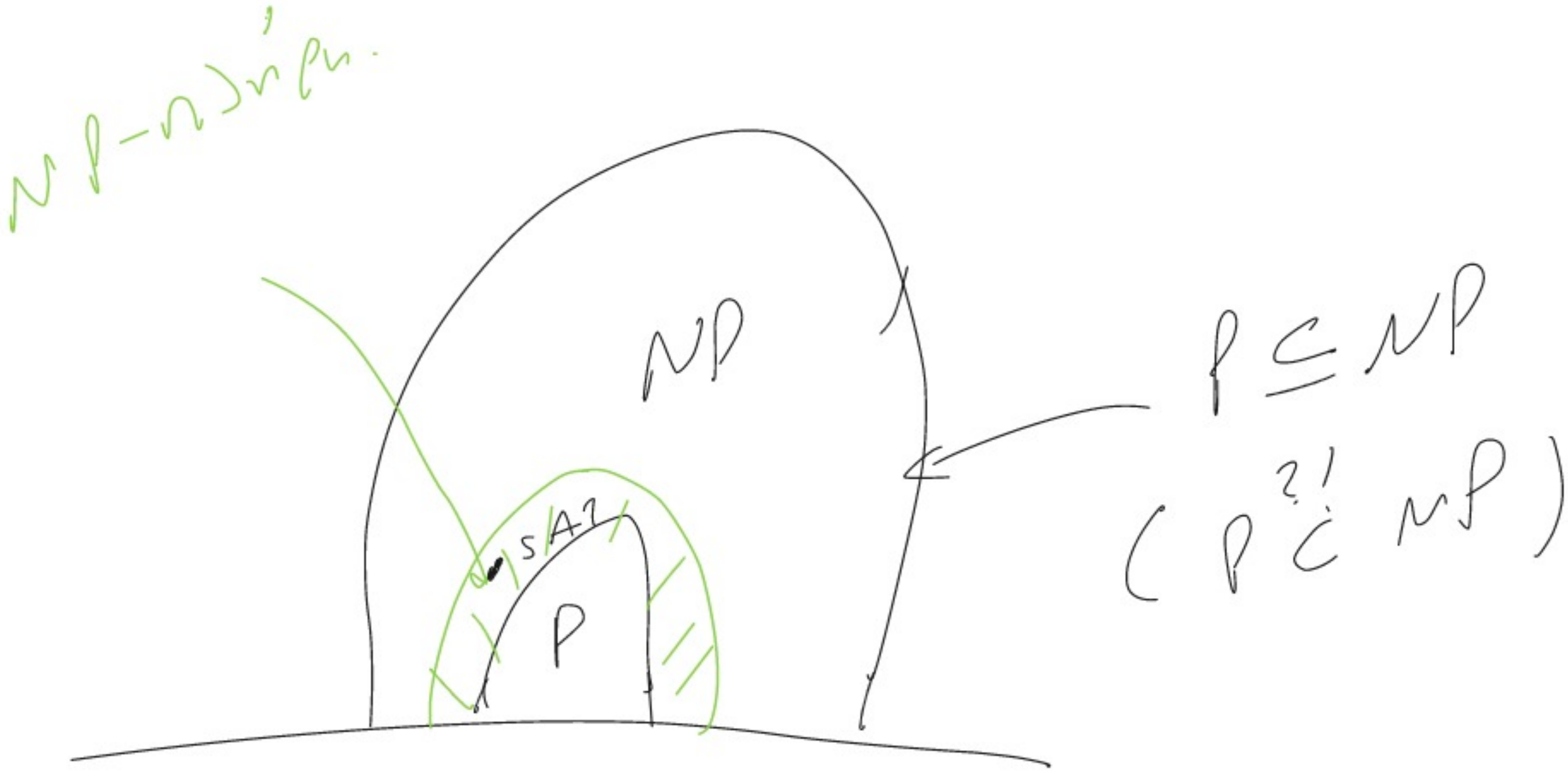
$v = [T \quad T \quad F \quad T]$

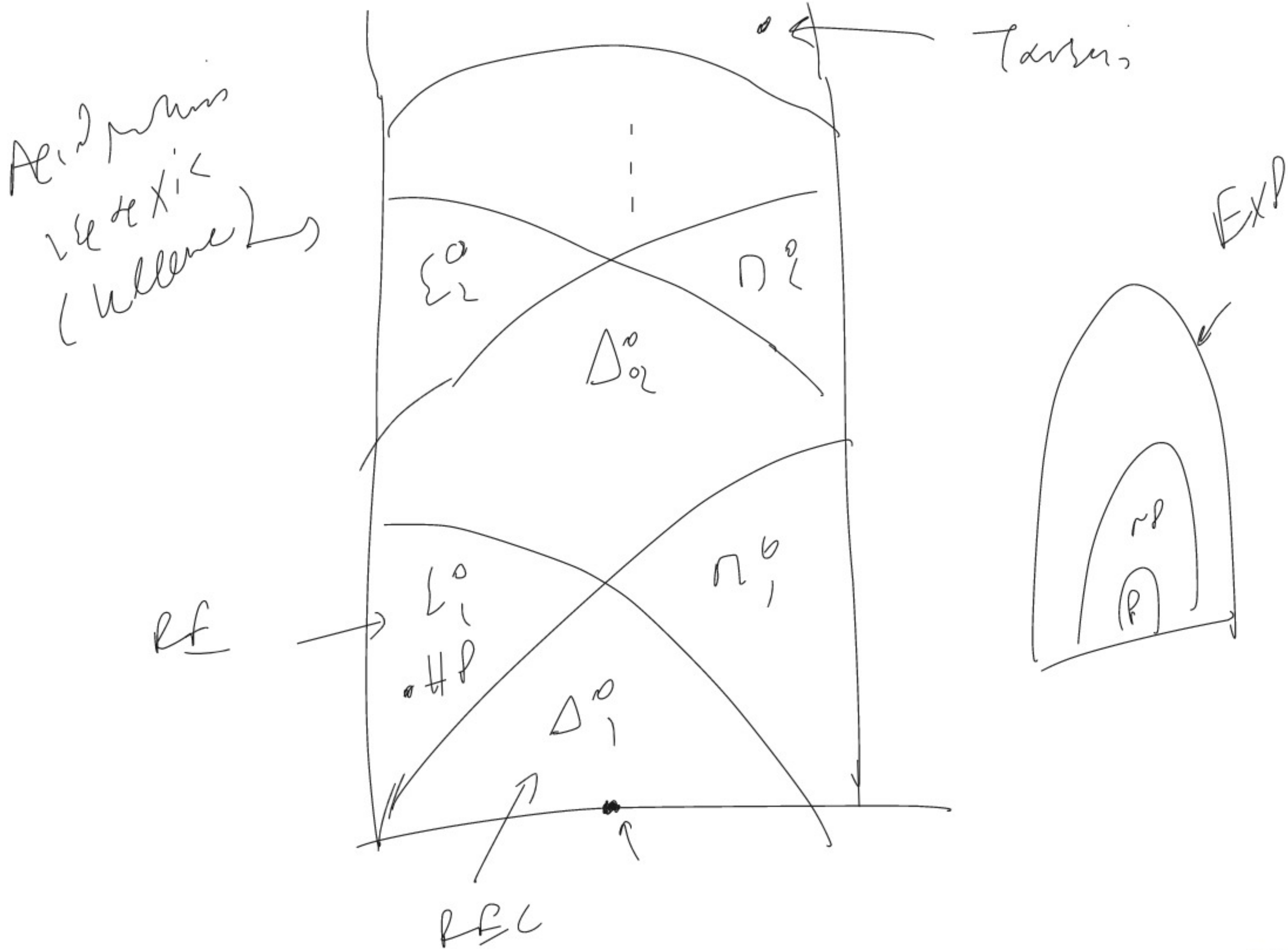
$v = [T \quad T \quad F \quad ?]$

$$P \overset{?!}{=} NP \qquad (P \subseteq NP)$$

προβλήματα με πολ/κό αλγόριθμο

προβλήματα που επαληθεύονται σε πολ/κό χρόνο

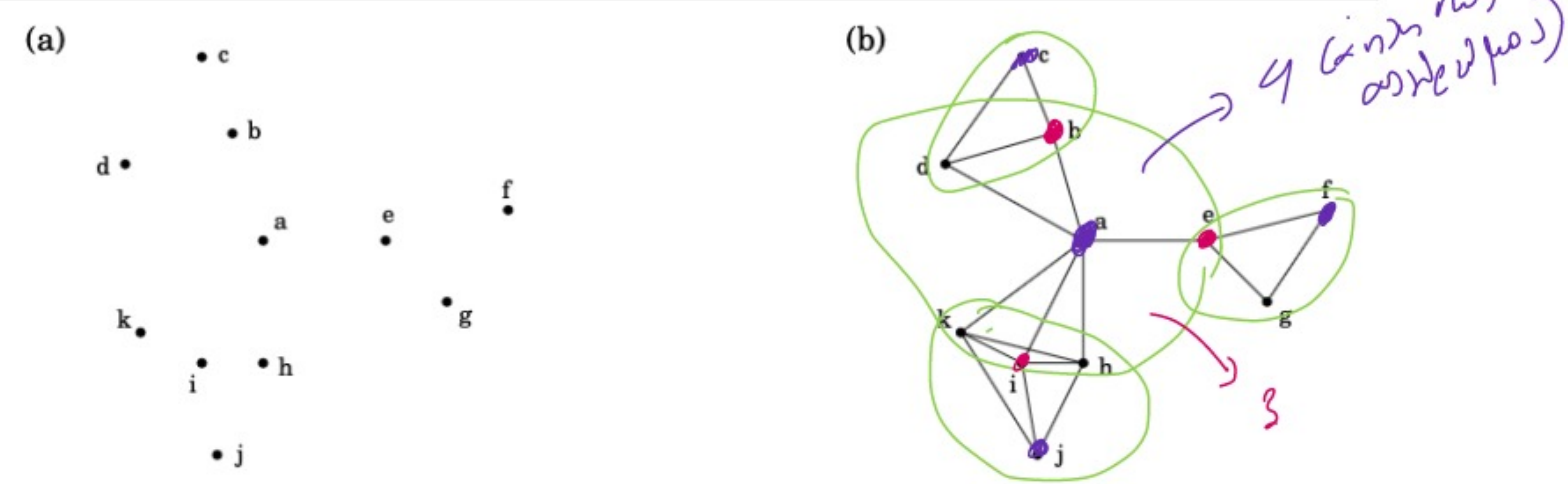

NP-complete
NP
SAT
P
P ⊆ NP
(P ⊊ NP ?!)



Σ^0_2
Π^0_2
Δ^0_2
Σ^0_1
Π^0_1
Δ^0_1
HP
EXP
NP
P

Κάλυψη Συνόλου

Figure 5.11 (a) Eleven towns. (b) Towns that are within 30 miles of each other.



Είσοδος: στοιχεία B, σύνολα $S_1, \dots, S_m \subseteq B$

Έξοδος: επιλογή S_i τ.ω. η ένωσή τους είναι το B

Κόστος: # των συνόλων που επιλέγει.

$OPT =$ βέλτιστο κάλυμμα

Η λύση του άπληστου αλγόριθμου είναι:

$$\log n \cdot OPT$$

n το πλήθος στοιχείων $\qquad n_t / OPT$

$n_t =$ ακάλυπτα στοιχεία μετά από t-επαναλήψεις

$$n_{t+1} \leq n_t - \frac{n_t}{OPT} = n_t \left(1 - \frac{1}{OPT}\right) \qquad (1-x) \leq e^{-x}$$

$$\vdots$$

$$n_t \leq n_0 \left(1 - \frac{1}{OPT}\right)^t < n_0 \cdot e^{-\frac{t}{OPT}} = n \cdot e^{-\frac{t}{OPT}}$$

Το n_t γίνεται 0 όταν $t = OPT \cdot \log n$ άρα

ο άπληστος αλγόριθμος θα επιλέξει $OPT \cdot \log n$

σύνολα.