

# Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσαρός

Διάλεξη 8η

# Περιεχόμενα

- **Ευρετήρια**

# Κυριτά πρότυπα προσπέλασης

- Τι γίνεται όταν το πρότυπο προσπέλασης στις εγγραφές δεν είναι ομοιόμορφο?
- Τα ευρετήρια που παρουσιάσαμε είναι κατάλληλα για ομοιόμορφα πρότυπα προσπέλασης
- $n$ : αριθμός εγγραφών
- $R_i$ :  $i$ -οστή εγγραφή
- $\Pr(R_i)$ : πιθανότητα προσπέλασης εγγραφής  $R_i$
- $I_{pb}(R_i)$ : αριθμός index nodes μέχρι να φτάσουμε στην εγγραφή ή index node  $R_i$
- $a_i$ : αναπαριστά έναν index node
- $d(a_i)$ : fanout του κόμβου  $a_i$
- $\text{Path}(R_i)$ : σύνολο των index nodes από τη ρίζα μέχρι την εγγραφή  $R_i$
- $f(a_i)$ : κόστος “ενεργοποίησης” index node  $a_i$ . Συνήθως  $f(a_i)=d(a_i)$ .

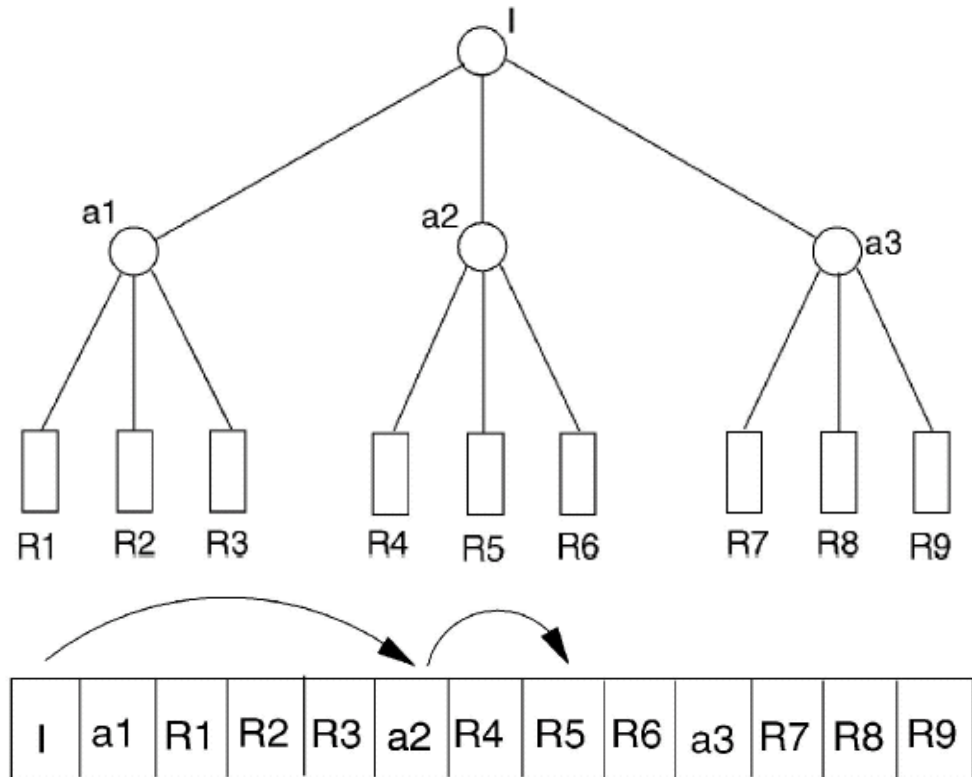
# Ισοζυγισμένα δενδρικά ευρετήρια

Έστω ότι το μέσο κόστος εντοπισμού μιας εγγραφής εκφάζεται ως:

$$\sum_{1 \leq i \leq n} Pr(R_i) \sum_{a_j \in Path(R_i)} f(a_j)$$

Το διπλανό ευρετήριο είναι βέλτιστο για ομοιόμορφη πιθανότητα προσπέλασης.

Μέσο κόστος:  $C(T_{d=3}^B) = 6$



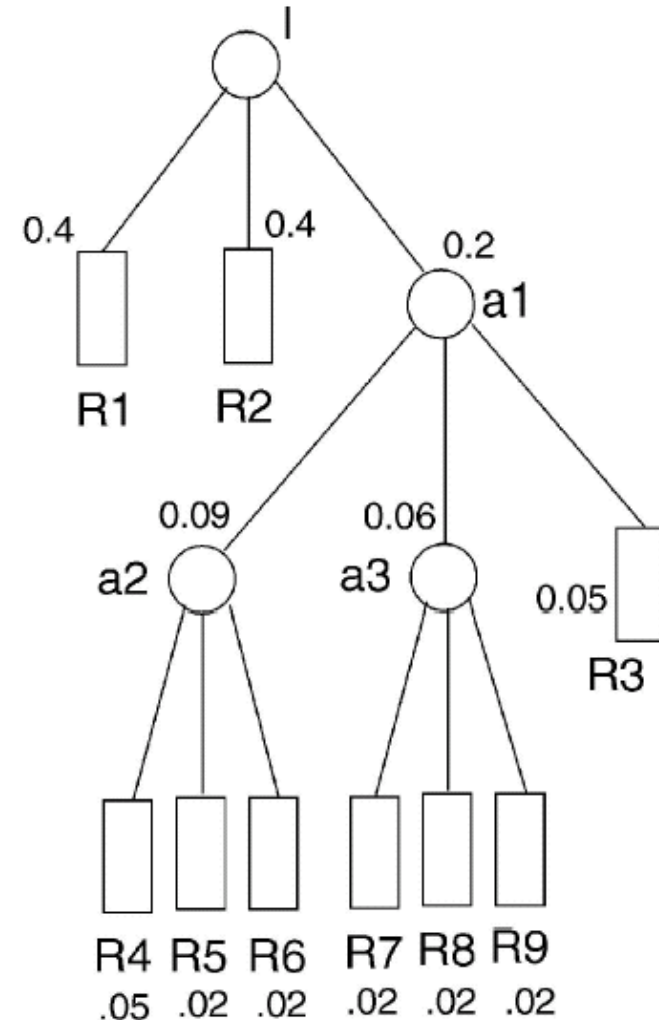
# Μη ισοζυγισμένα δενδρικά ευρετήρια σταθερού fanout [π.χ., d=3]

| Data record                  | R1  | R2  | R3   | R4   | R5   | R6   | R7   | R8   | R9   |
|------------------------------|-----|-----|------|------|------|------|------|------|------|
| $Pr(R_i)$                    | 0.4 | 0.4 | 0.05 | 0.05 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| $I_{pb}(R_i)$ in $T_{d=3}^B$ | 2   | 2   | 2    | 2    | 2    | 2    | 2    | 2    | 2    |
| $I_{pb}(R_i)$ in $T_{d=3}^I$ | 1   | 1   | 2    | 3    | 3    | 3    | 3    | 3    | 3    |

$$\begin{aligned}
 C(T_{d=3}^I) &= 0.4*3+0.4*3+ && \text{[για } R_1 \text{ και } R_2\text{]} \\
 &0.05*(3+3)+ && \text{[για } R_3\text{]} \\
 &0.05*(3+3+3)+ && \text{[για } R_4\text{]} \\
 &5*0.02*(3+3+3) && \text{[για υπόλοιπες]} \\
 &= 0.4*6+0.05*15+45*0.02 = \mathbf{4.28}
 \end{aligned}$$

Σειρά εκπομπής στο ασύρματο κανάλι

|   |    |    |    |    |    |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|----|----|----|----|----|
| I | R1 | R2 | a1 | R3 | a2 | R4 | R5 | R6 | a3 | R7 | R8 | R9 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|



# Αλγόριθμος κατασκευής $T^I_{d=j}$

Algorithm CF: Use access frequencies to build an index tree with a fixed fanout  $d$ .

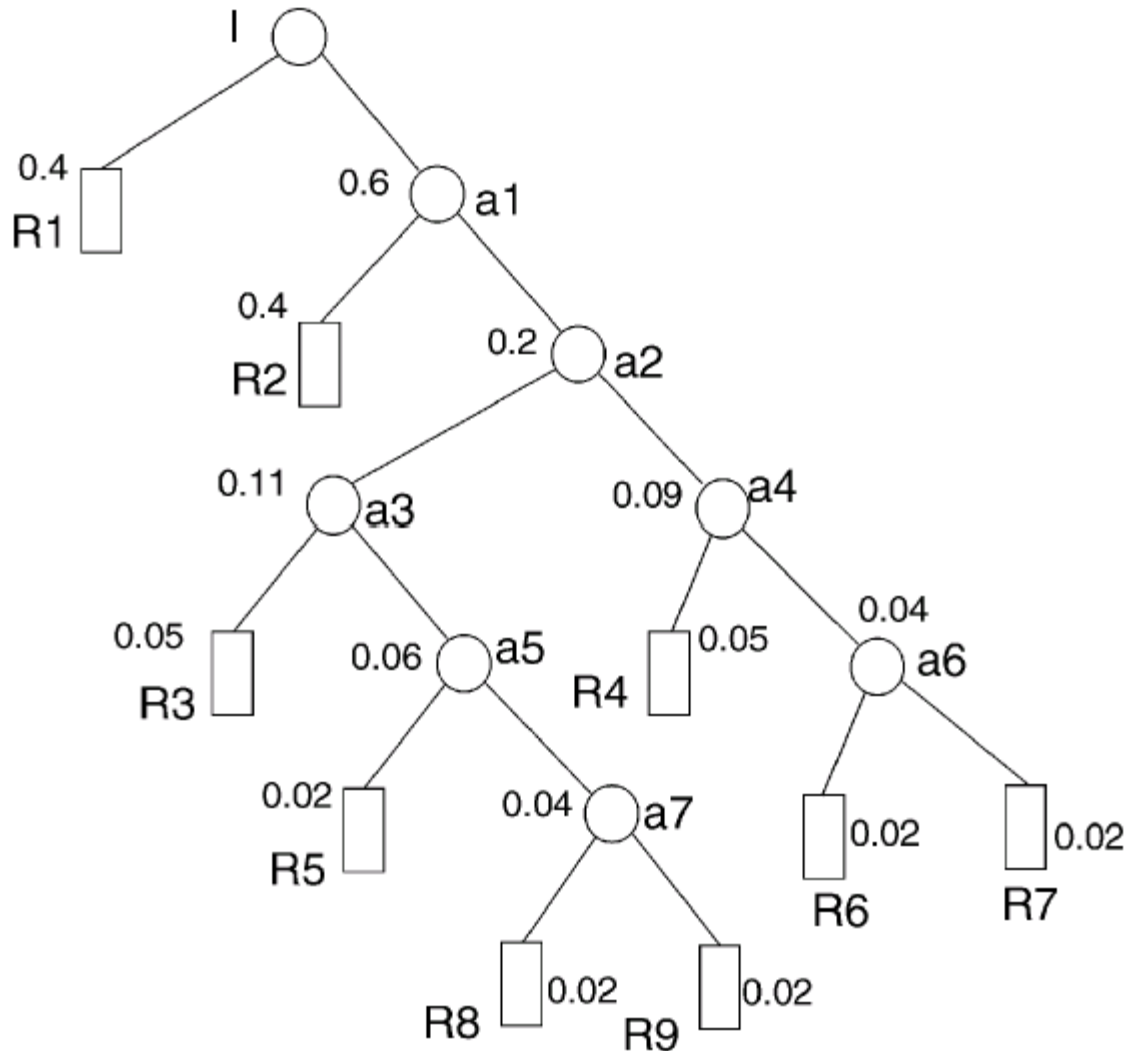
**Step 1:** Initially, we have a forest of  $n$  subtrees, each of which is a single node labeled with the corresponding access frequency.

**Step 2:** Attach the  $d$  subtrees with the smallest labels to a new node. Label the resulting subtree with the sum of all labels from its  $d$  child subtrees.

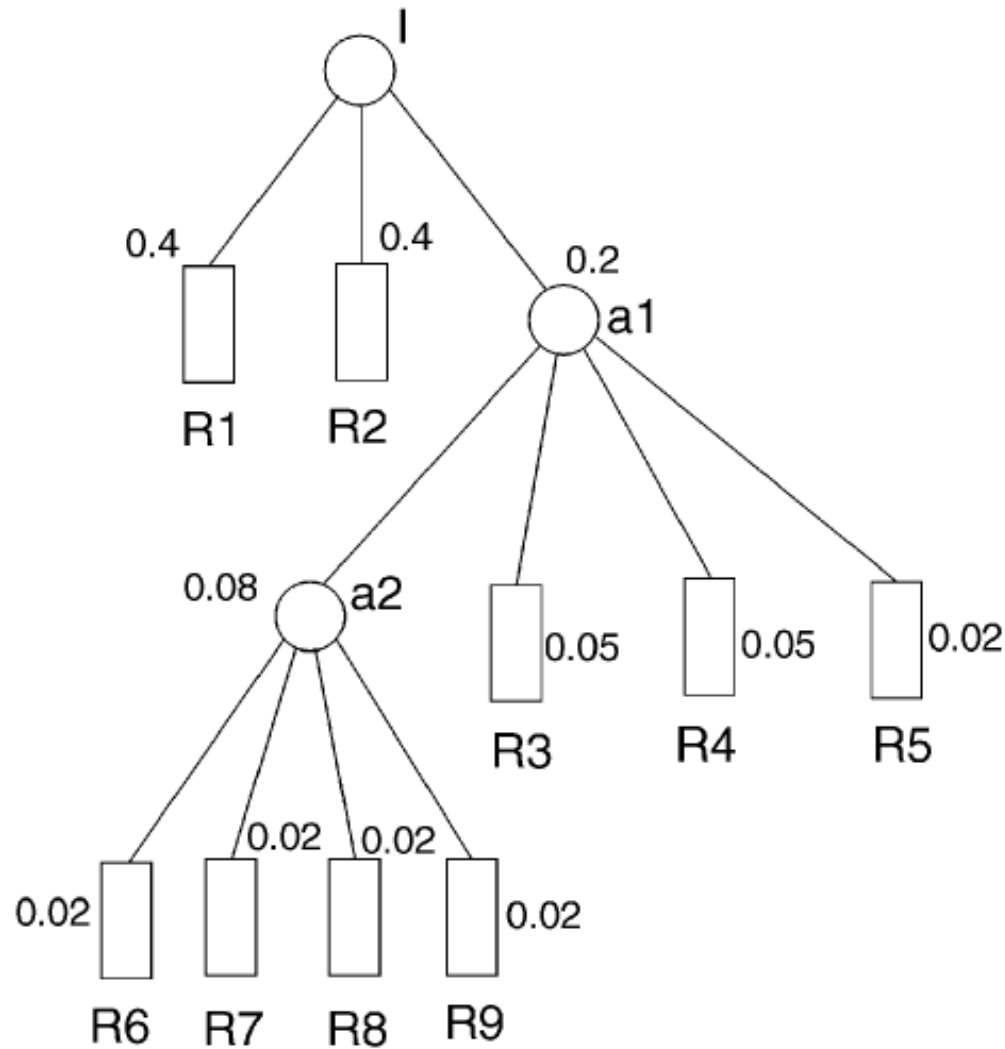
**Step 3:**  $n = n - d + 1$ . (Then,  $n$  is the number of remaining subtrees.) If  $n = 1$  stop else goto Step 2.

**Κατασκευάζει ένα Huffman δένδρο με fanout =  $d$ .**  
**Στο βασικό Huffman δένδρο, fanout = 2.**

# Μη ισοζυγισμένα δενδρικά ευρετήρια σταθερού fanout [π.χ., $d=2$ ]



# Μη ισοζυγισμένα δενδρικά ευρετήρια σταθερού fanout [π.χ., $d=4$ ]





# Μη ισοζυγισμένα δένδρα σταθερού fanout

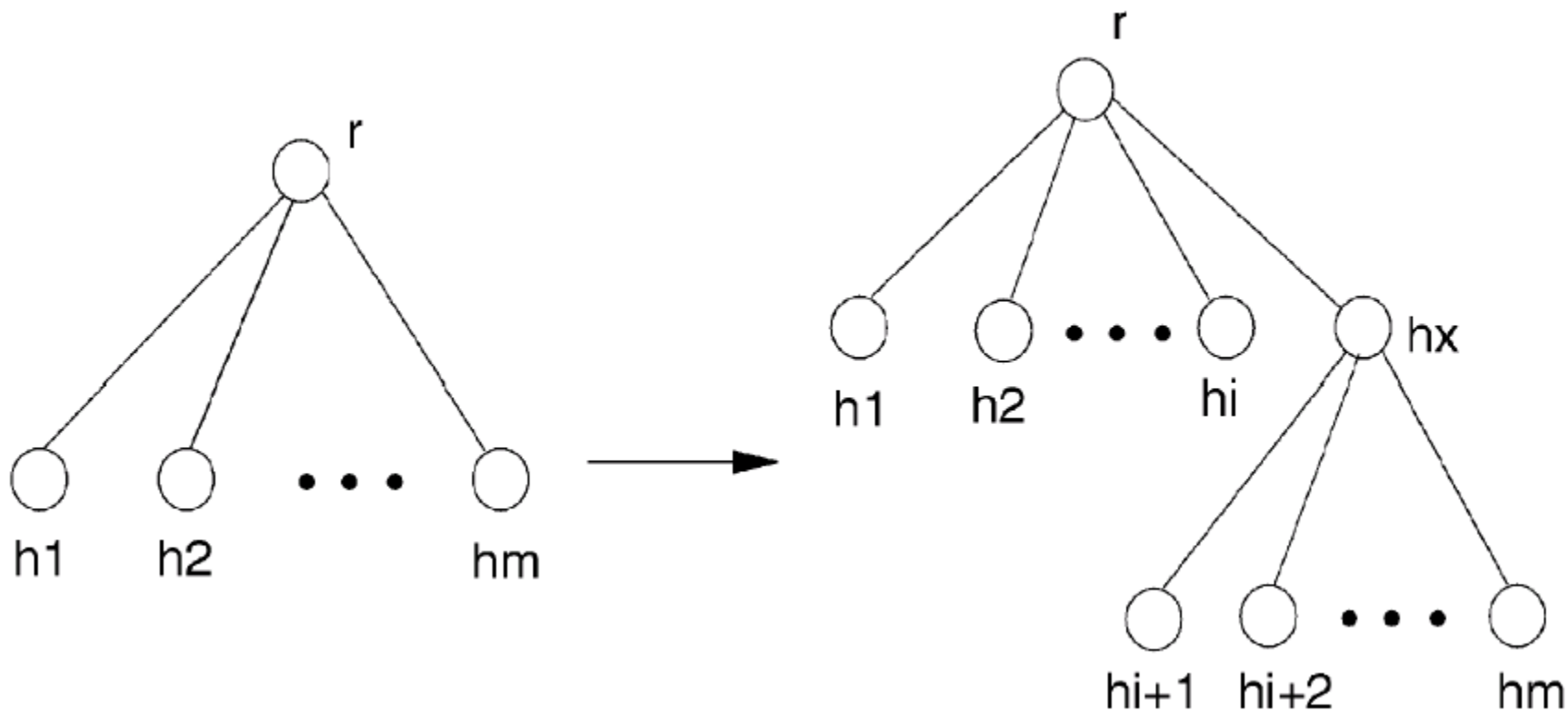
- Δεν υπάρχει μονοτονική σχέση, ούτε αύξουσα ούτε φθίνουσα, για τις τιμές κόστους  $C(T_{d=j}^I)$  σε σχέση με το επιτρεπτό fanout  $j$ .
- Αυτό το γεγονός, σε συνδυασμό με την ιεραρχική φύση της κατασκευής, υπονοεί ότι, επιτρέποντας μεταβλητά fanouts στους κόμβους του δένδρου, μπορούμε να ελαττώσουμε ακόμα περισσότερο το μέσο κόστος εντοπισμού των εγγραφών

# Βασική ιδέα (1/2)

**Lemma 1.** *Suppose that node  $r$  has  $m$  child nodes,  $h_1, h_2, \dots,$  and  $h_m$ , which are sorted according to descending order of  $Pr(h_j)$ ,  $1 \leq j \leq m$ , i.e.,  $Pr(h_j) \geq Pr(h_k)$  if and only if  $j \leq k$ . Then, the average cost of index probes can be reduced by grouping nodes  $h_{i+1}, h_{i+2}, \dots,$  and  $h_m$  and attaching them under a new child node if and only if*

$$(m - i - 1) \sum_{1 \leq j \leq i} Pr(h_j) > \sum_{i+1 \leq j \leq m} Pr(h_j).$$

# Βασική ιδέα (2/2)



# Μη ισοζυγισμένα δένδρα μεταβλητού fanout $T^I_V$ : Αλγόριθμος κατασκευής

12

Algorithm VF:

**Step 1:** Assume that  $R_1, R_2, \dots,$  and  $R_n$  have been sorted according to descending order of  $Pr(R_j), 1 \leq j \leq n,$  i.e.,  
 $Pr(R_j) \geq Pr(R_k)$  iff.  $j \leq k.$

**Step 2:** *Partition*  $(R_1, R_2, \dots, R_n).$

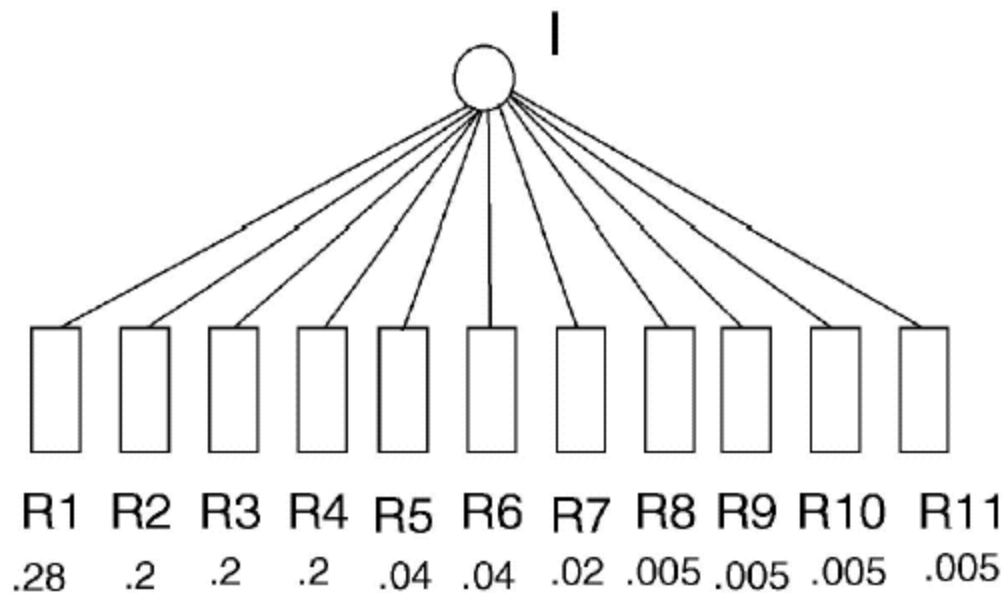
**Step 3:** Report the resulting index tree.

# Μη ισοζυγισμένα δένδρα μεταβλητού fanout $T_V^I$ : Αλγόριθμος κατασκευής

Procedure *Partition* ( $h_1, h_2, \dots, h_m$ ):

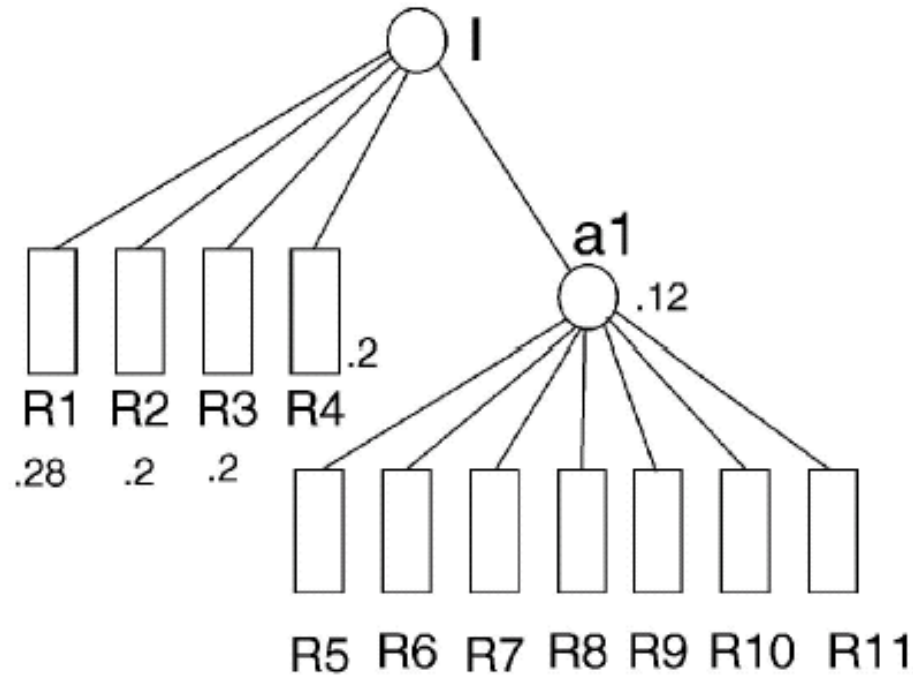
1. Let  $y(i) = (m - i - 1) \sum_{1 \leq j \leq i} Pr(h_j) - \sum_{i+1 \leq j \leq m} Pr(h_j)$ .  
Determine  $i^*$  such that  $y(i^*) = \max_{\forall i \in \{1, m-2\}} \{y(i)\}$ .
2. If  $y(i^*) \leq 0$ , then return.
3. Attach nodes  $h_{i^*+1}, h_{i^*+2}, \dots, h_m$  under a new index node  $hx$  in the index tree.
4. *Partition* ( $h_{i^*+1}, h_{i^*+2}, \dots, h_m$ ).
5. Insert  $hx$  into the ordered list ( $h_1, h_2, \dots, h_{i^*}$ ) and relabel them as ( $h_1, h_2, \dots, h_{i^*+1}$ ) according to descending order of  $Pr(h_j)$ ,  $1 \leq j \leq i^* + 1$ .
6. *Partition* ( $h_1, h_2, \dots, h_{i^*+1}$ ).
7. Return.

# Παράδειγμα κατασκευής $T_V^I$ (1/6)



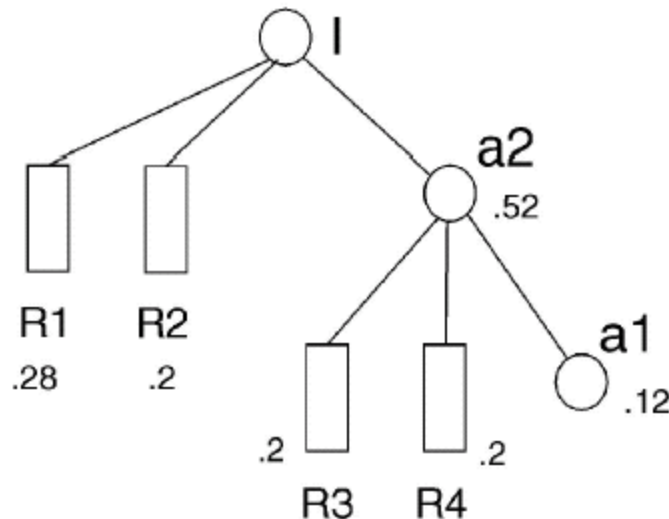
| $i$                                 | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8       | 9    |
|-------------------------------------|--------|--------|--------|--------|--------|--------|--------|---------|------|
| $(10 - i) \sum_{1 < j < i} Pr(h_j)$ | 9*.208 | 8*0.48 | 7*0.68 | 6*0.88 | 5*0.92 | 4*0.96 | 3*0.98 | 2*0.985 | 0.99 |
| $\sum_{i+1 < j < 11} Pr(h_j)$       | 0.72   | 0.52   | 0.32   | 0.12   | 0.08   | 0.04   | 0.02   | 0.015   | 0.01 |
| $y(i)$                              | 1.8    | 3.32   | 4.44   | 5.16   | 4.52   | 3.8    | 2.92   | 1.955   | 0.98 |

# Παράδειγμα κατασκευής $T_V^I$ (2/6)



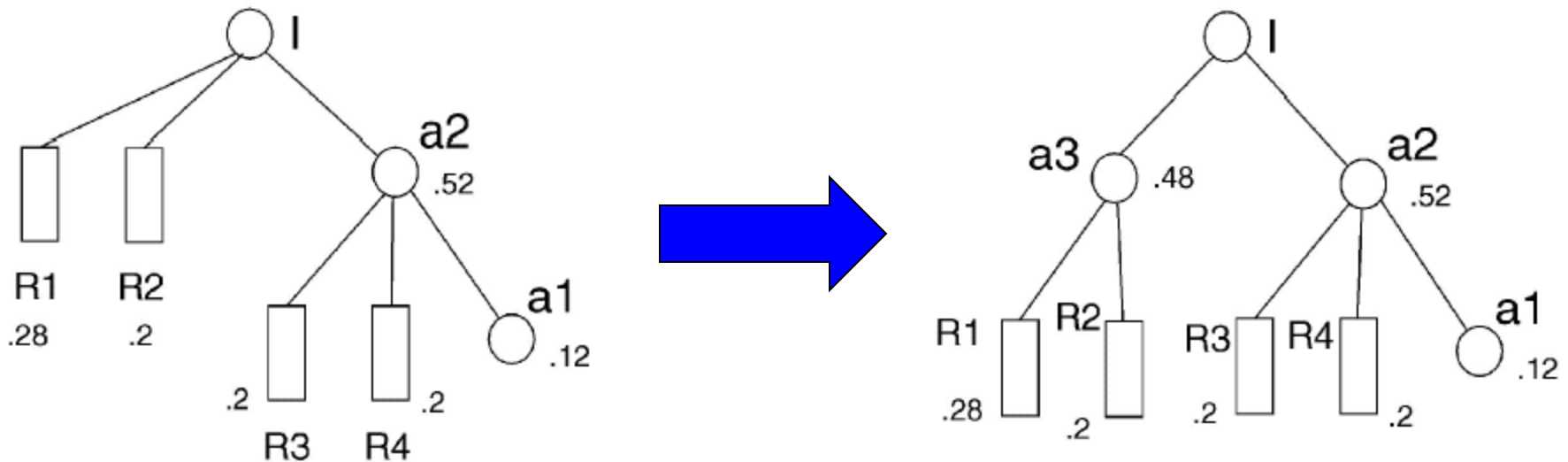
# Παράδειγμα κατασκευής $T_V^I$ (3/6)

| $i$                                | 1          | 2          | 3    |
|------------------------------------|------------|------------|------|
| $(4 - i) \sum_{1 < j < i} Pr(h_j)$ | $3 * 0.28$ | $2 * 0.48$ | 0.68 |
| $\sum_{i+1 < j < 5} Pr(h_j)$       | 0.72       | 0.52       | 0.32 |
| $y(i)$                             | 0.12       | 0.44       | 0.32 |

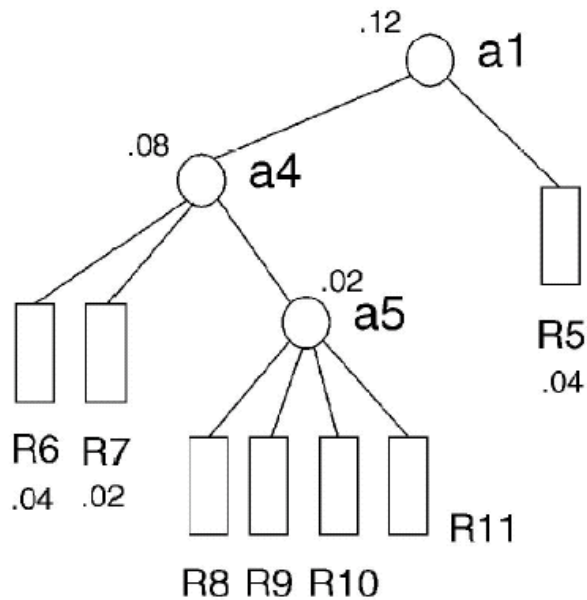
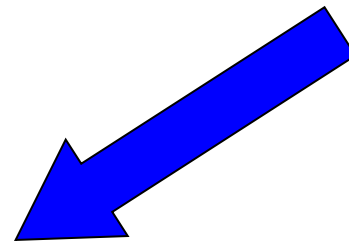
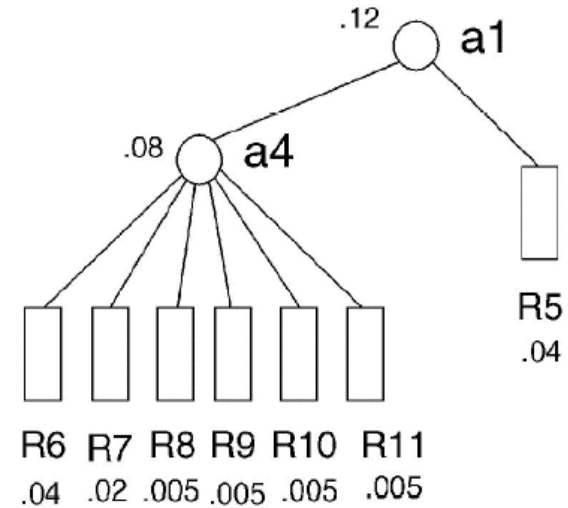
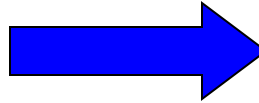
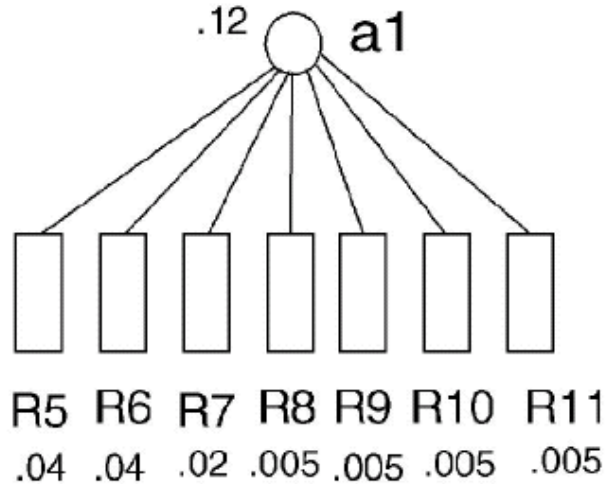




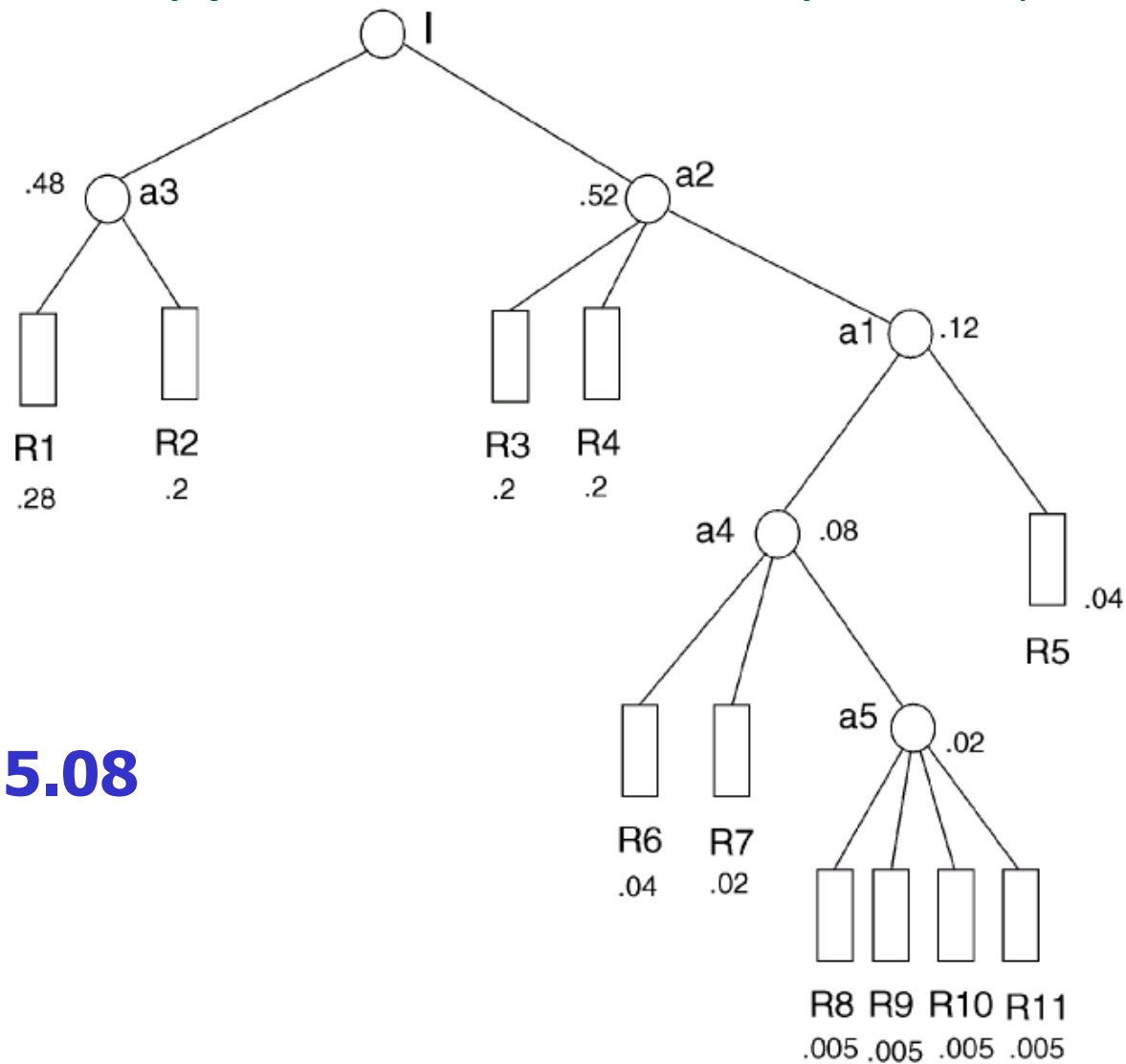
# Παράδειγμα κατασκευής $T_V^I$ (4/6)



# Παράδειγμα κατασκευής $T_V^I$ (5/6)

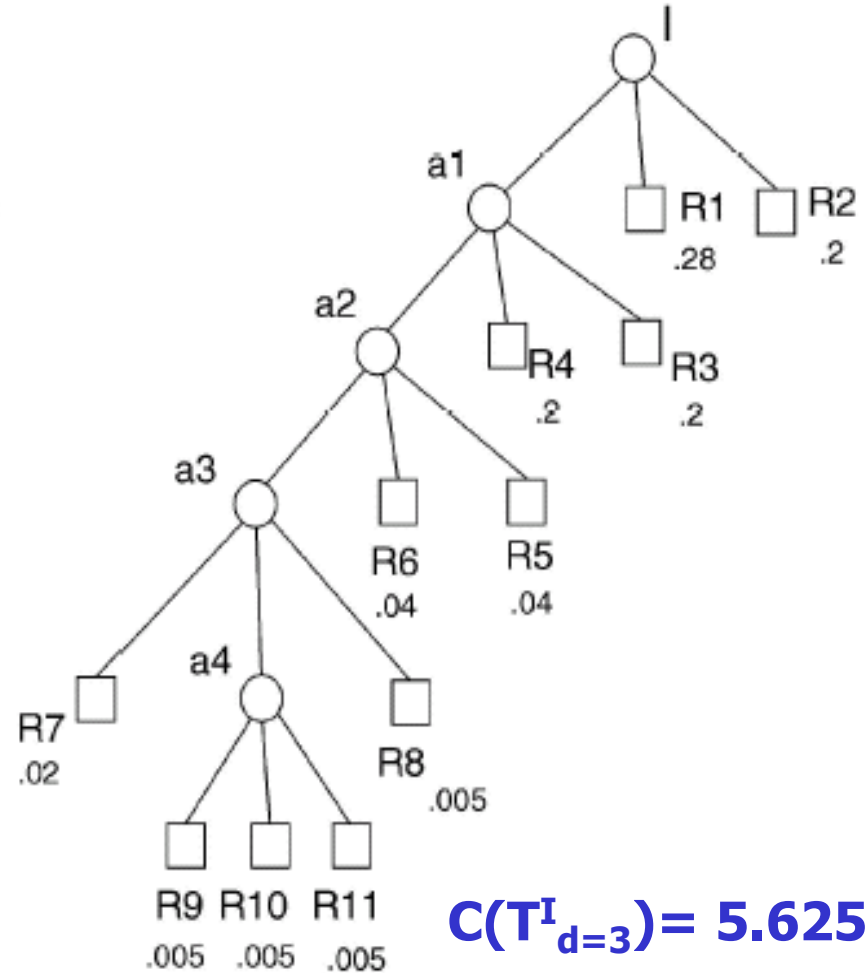
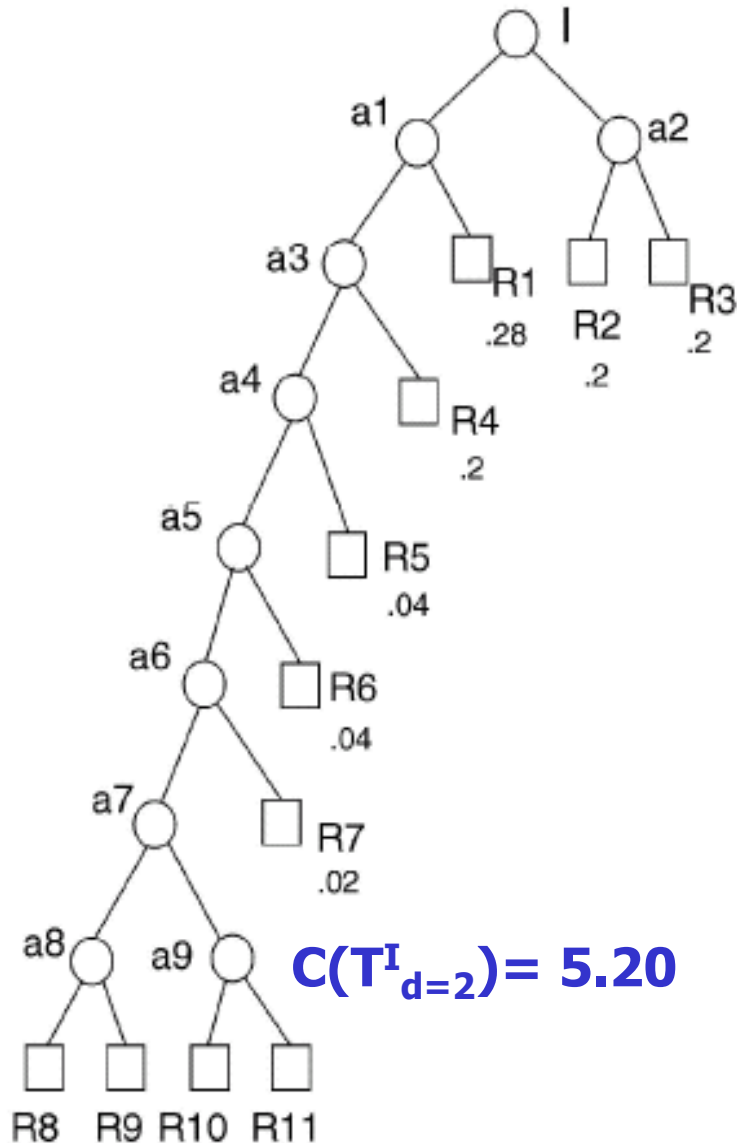


# Παράδειγμα κατασκευής $T_V^I$ (6/6)



$$C(T_V^I) = 5.08$$

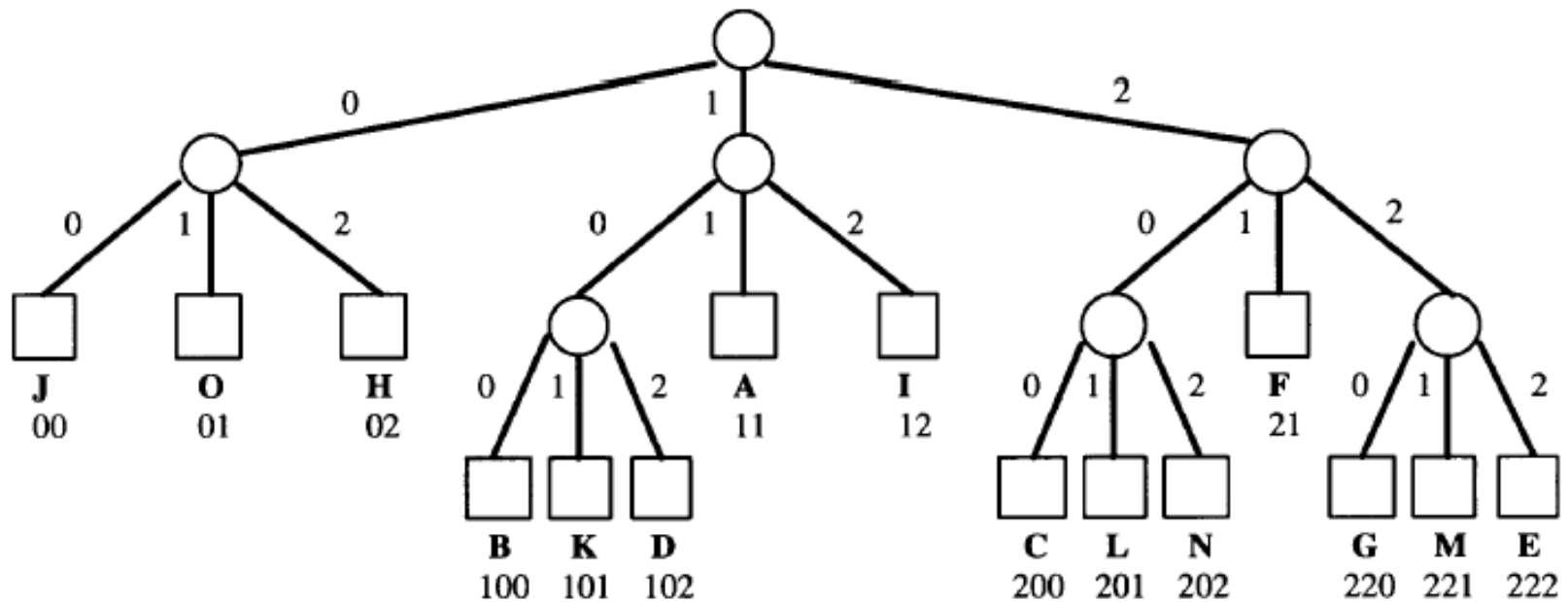
# Αντιδιαστολή $T^I_V$ με $T^I_{d=2}$ και $T^I_{d=3}$



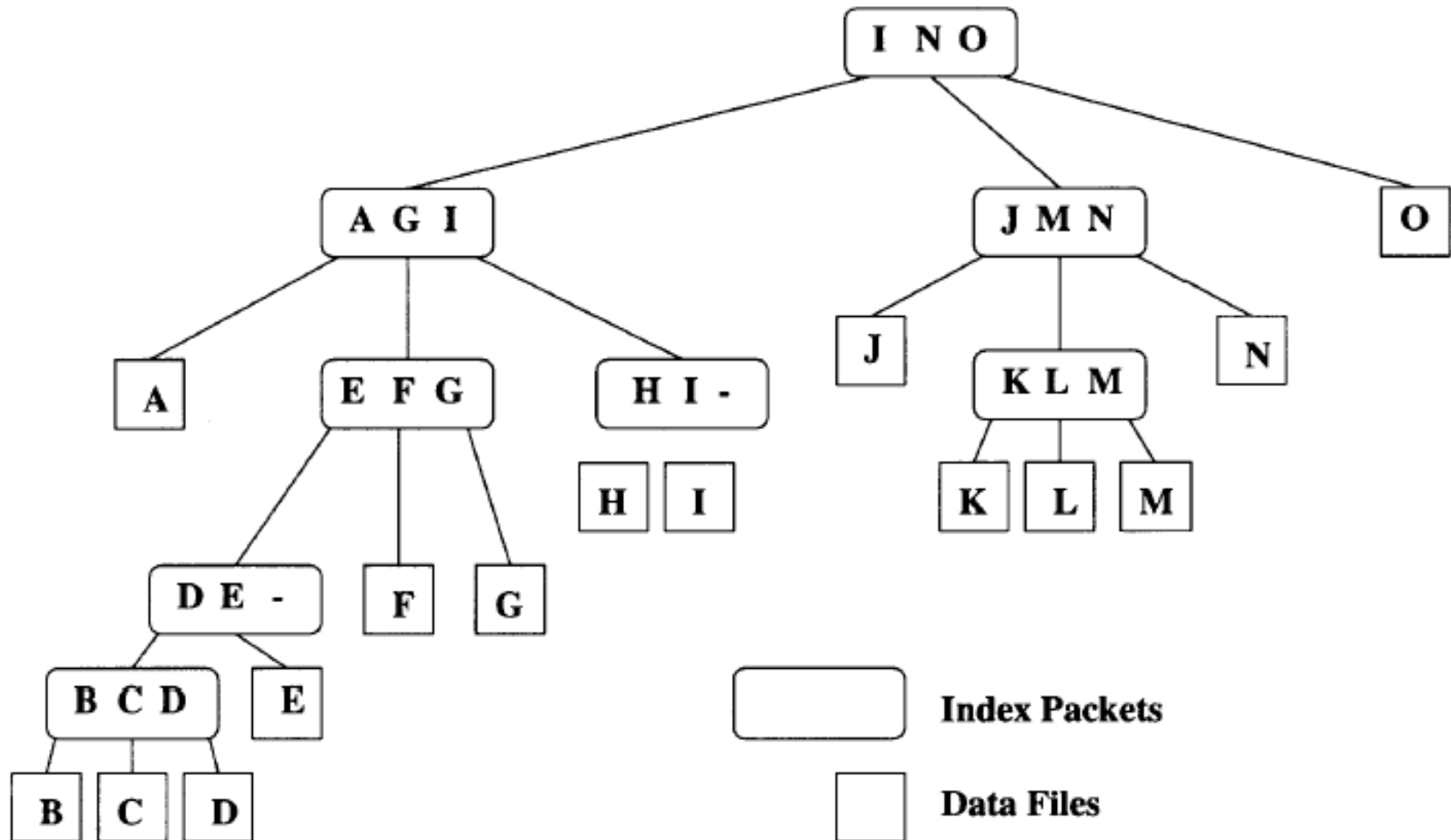
# “Αλφαβητικά” δενδρικά ευρετήρια

- Ομοιόμορφη προσπέλαση & ταξινομημένα κλειδιά
  - (1,m) indexing
  - Distributed indexing
  - Exponential indexing
- Μη ομοιόμορφη προσπέλαση & μη ταξινομημένα κλειδιά
  - $T_{d=j}^I$
  - $T_V^I$
- Μη ομοιόμορφη προσπέλαση & ταξινομημένα κλειδιά ????

# Το πρόβλημα των Huffman δένδρων



# Τα Alphabetic δένδρα

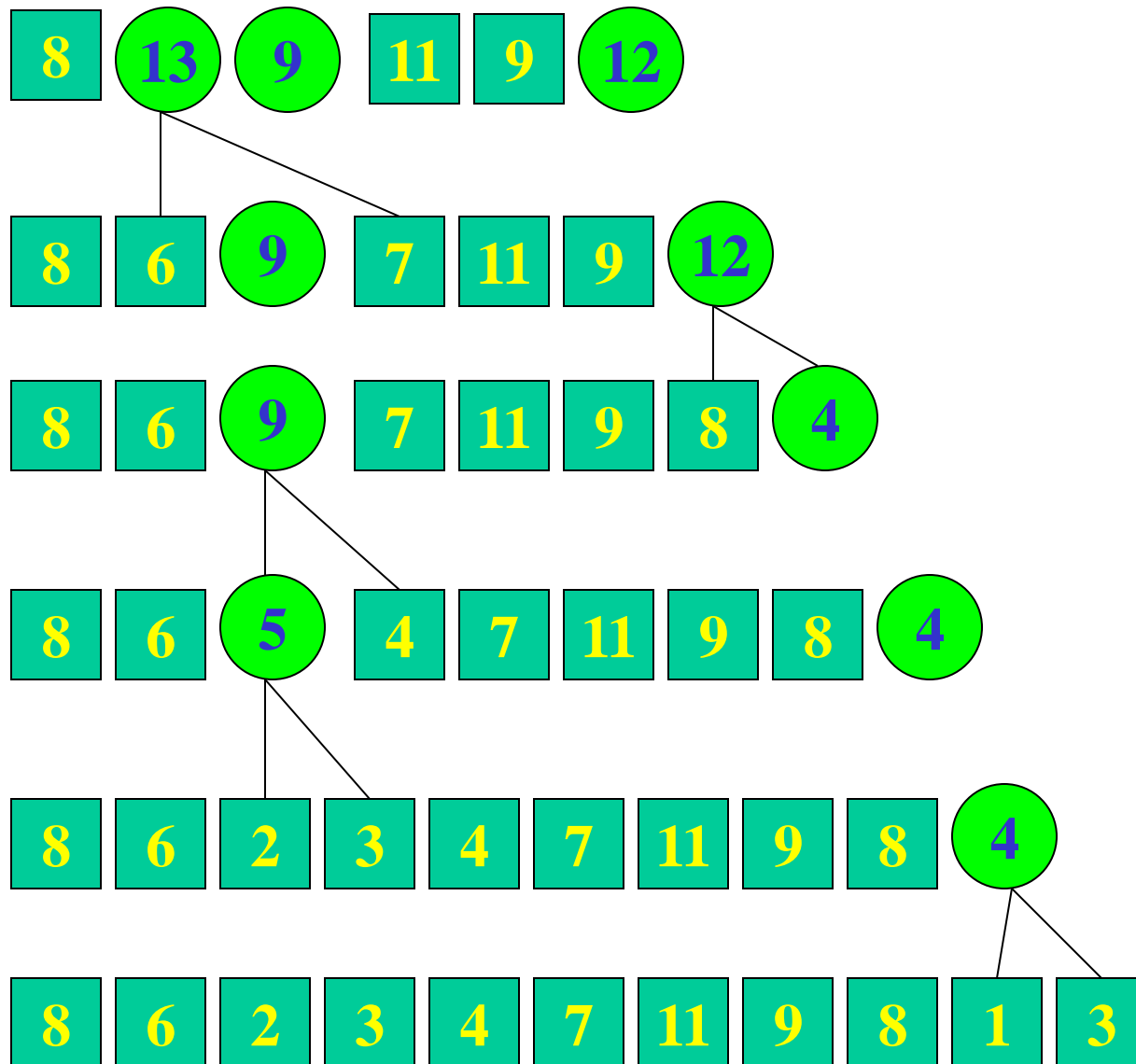


# Παράδειγμα κατασκευής δυαδικού αλφαβητικού δένδρου

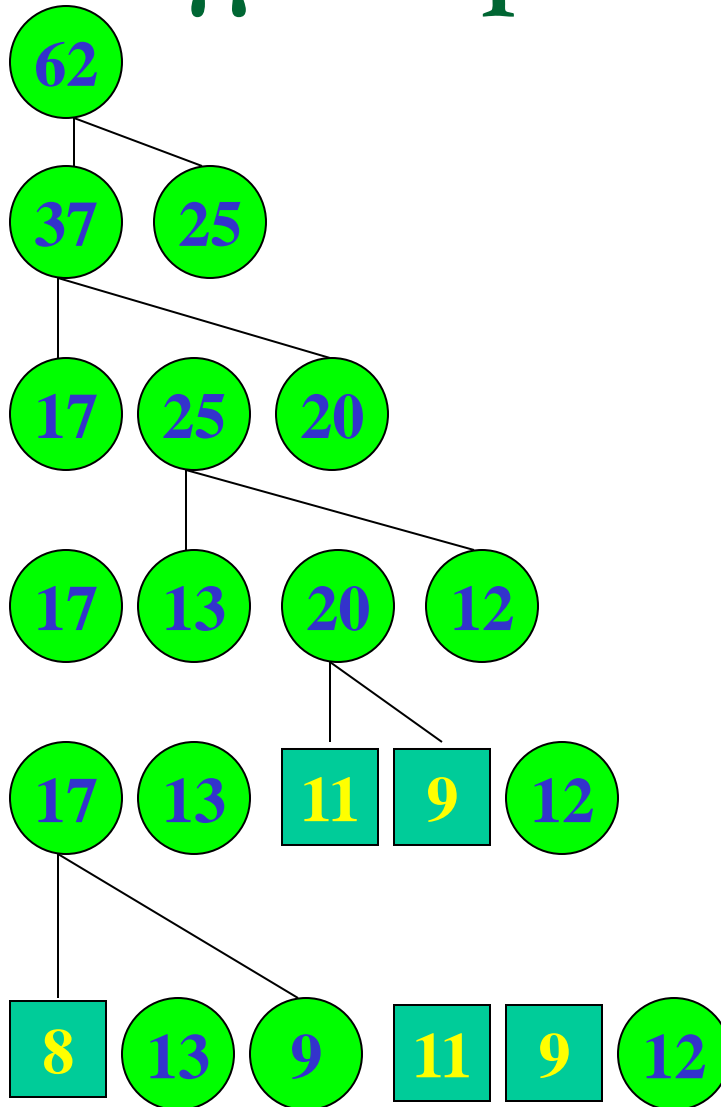
|   |   |   |   |   |   |    |   |   |   |   |
|---|---|---|---|---|---|----|---|---|---|---|
| A | B | C | D | E | F | G  | H | I | J | K |
| 8 | 6 | 2 | 3 | 4 | 7 | 11 | 9 | 8 | 1 | 3 |



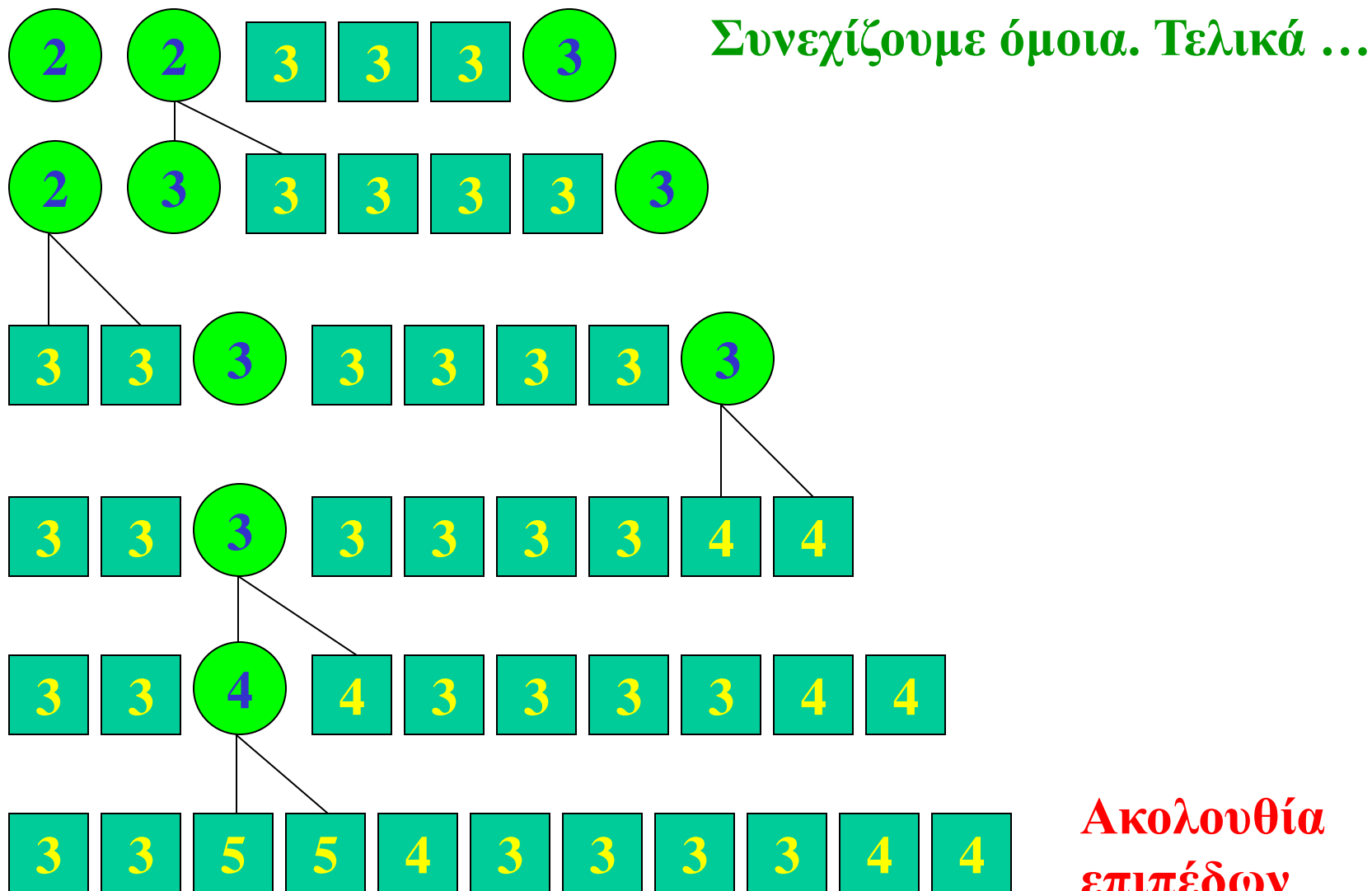
# Παράδειγμα Alphabetic tree (1/4)



# Παράδειγμα Alphabetic tree (2/4)



# Παράδειγμα Alphabetic tree (3/4)



# Alphabetic tree (4/4)

