BINARY SEARCH (ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ)

ΕΙΣΟΔΟΣ: Ταξινομημένος Πίνακας A, Στοιχείο x
ΕΞΟΔΟΣ:
Αν το στοιχείο υπάρχει στον πίνακα:
- Επιστρέφεται η θέση του.

Αν το στοιχείο δεν υπάρχει στον πίνακα:
- Επιστρέφεται 0

ΨΕΥΔΟΚΩΔΙΚΑΣ (αναδρομική υλοποίηση)

```
procedure BinarySearch(A,x,start,finish)

if start>finish then
  return 0
else
  middle=(start+finish) div 2
  if (x==A[middle]) then
    return middle
  else if (x<A[middle]) then
    pos=BinarySearch(A,x,start,middle-1)
    return pos
  else if (x>A[middle]) then
    pos=BinarySearch(A,x,middle+1,finish)
    return pos
  end if
end if

end procedure
```

ΑΛΓΟΡΙΘΜΙΚΗ ΙΔΕΑ: Εξετάζεται το μεσαίο στοιχείο:
Αν x = μεσαίο, επιστρέφεται η θέση.
Αν x < μεσαίο, ψάχνουμε στο αριστερό μισό του πίνακα
Αν x > μεσαίο, ψάχνουμε στο δεξί μισό του πίνακα

ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ:

Εκτελούμε τον αλγόριθμο ψάχνοντας το στοιχείο 11 στον πίνακα:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	21	23	27	31	33	37	41	43

Κλήση: BinarySearch(A,11,1,15): middle=(1+15) div 2=8. x<A[middle]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	21	23	27	31	33	37	41	43
start														finish

Αναδρομική Κλήση: BinarySearch(A,11,1,7) : middle=(1+7) div 2=4 x>A[middle]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	21	23	27	31	33	37	41	43
start						finish								

Αναδρομική Κλήση: BinarySearch(A,11,5,7) : middle=(5+7) div 2=6 x<A[middle]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	21	23	27	31	33	37	41	43
				start		finish								

Αναδρομική Κλήση: BinarySearch(A,11,5,5) : middle=(5+5) div 2=5 x=A[middle]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	21	23	27	31	33	37	41	43
				start=finish										

ΠΟΛΥΠΛΟΚΟΤΗΤΑ:
Στην χειρότερη περίπτωση, μία αναδρομική κλήση για τα «μισά» δεδομένα, άρα:

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$$

Β΄Θεωρημα .... Κυριαρχίας

$$T(n) = \Theta(\log n)$$

Καλύτερη Περίπτωση: Θ(1)

MERGESORT (ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΣΥΓΧΩΝΕΥΣΗ)

ΕΙΣΟΔΟΣ: Πίνακας Στοιχείων
ΕΞΟΔΟΣ: Ταξινομημένος (σε αύξουσα σειρά) πίνακας

ΨΕΥΔΟΚΩΔΙΚΑΣ

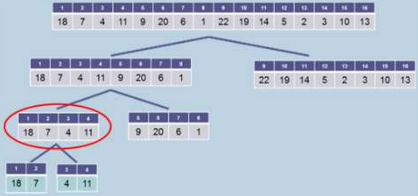
```
procedure MergeSort(A,start, finish)
   if |A|<=2 then
      Ταξινόμησε τον A
   else
      middle=(start+finish) div 2
      A_1=MergeSort(A,start,middle)
      A_2=MergeSort(A,middle+1,finish)
      A=Merge(A_1,A_2)
   end if
end procedure
```

```
procedure Merge(A,B)
   i=1, j=1, k=1
   while (i<=n AND j<=m)
      if (a_i<b_j) then c_k=a_i ; i=i+1
      else c_k=b_j ; j=j+1
      k=k+1
   end while
   Όσα στοιχεία του A ή του B περίσσεψαν
         τα βάζουμε στο τέλος του C
   return C
end procedure
```

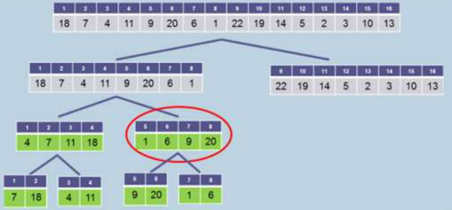
ΑΛΓΟΡΙΘΜΙΚΗ ΙΔΕΑ: Αναδρομικά:
Ταξινόμησε τον Αριστερό Υποπίνακα
Ταξινόμησε τον Δεξιό Υποπίνακα
Συγχώνευσε τους δύο σε έναν ενιαίο πίνακα

ΜΕΡΟΣ ΕΚΤΕΛΕΣΗΣ:

Ανάπτυξη Υποπινάκων:



Συγχώνευση Υποπινάκων:



ΠΟΛΥΠΛΟΚΟΤΗΤΑ:
Γίνονται δύο αναδρομικές κλήσεις για τα μισα δεδομένα. Ο χρόνος της συγχώνευσης (merge) είναι Θ(n).

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

Β' Θεωρημα .... Κυριαρχίας

$$T(n) = \Theta(n \log n)$$

Το Θεώρημα Κυριαρχίας: Έστω η αναδρομική εξίσωση $\mathbf{T(n) = aT\left(\frac{n}{b}\right) + f(n)}$ όπου a≥1, b>1 είναι σταθερές, και $f(n)$ είναι μια ασυμπτωτικά θετική συνάρτηση. Τότε διακρίνονται οι ακόλουθες τρεις περιπτώσεις:

Α) Αν $f(n) = O\left(n^{\log_b a-\varepsilon}\right)$ για κάποια σταθερά ε>0, τότε: $T(n) = \Theta\left(n^{\log_b a}\right)$

Β) Αν $f(n) = \Theta\left(n^{\log_b a}\right)$ τότε: $T(n) = \Theta\left(n^{\log_b a}\log n\right)$

Γ) Αν $f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right)$ για κάποια σταθερά ε>0 και $af\left(\frac{n}{b}\right) \le cf(n)$ για κάποια σταθερά c<1, τότε: $T(n) = \Theta(f(n))$

Παραδείγματα:

$$T(n) = 8T\left(\frac{n}{2}\right) + n$$

Έχω: $a=8,\quad b=2,\quad f(n)=n,\quad \log_b a = \log_2 8 = 3$

Ισχύει: $f(n) = n = O(n^{3-\varepsilon})$ για κάποια σταθερά ε>0

Άρα από την Α' περίπτωση του Θεωρήματος Κυριαρχίας έπεται ότι: $T(n)=\Theta(n^3)$

$$T(n) = 9T\left(\frac{n}{3}\right) + n^2$$

Έχω: $a=9,\quad b=3,\quad f(n)=n^2,\quad \log_b a = \log_3 9 = 2$

Ισχύει: $f(n) = n^2 = \Theta(n^2)$

Άρα από την Β' περίπτωση του Θεωρήματος Κυριαρχίας έπεται ότι: $T(n)=\Theta(n^2\log n)$

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

Έχω: $a=4,\quad b=2,\quad f(n)=n^3,\quad \log_b a = \log_2 4 = 2$

Ισχύει: $f(n) = n^3 = \Omega(n^{2+\varepsilon})$ για κάποια σταθερά ε>0

Ελέγχω αν υπάρχει c<1 τέτοιο ώστε:

$$af\left(\frac{n}{b}\right) \le cf(n) \Leftrightarrow 4f\left(\frac{n}{2}\right) \le cf(n) \Leftrightarrow 4\left(\frac{n}{2}\right)^3 \le cn^3 \Leftrightarrow 4\frac{n^3}{2^3} \le cn^3 \Leftrightarrow \frac{4}{8} \le c \Leftrightarrow \frac{1}{2} \le c$$

Άρα ισχύει για ½≤c<1.

Άρα από την Γ' περίπτωση του Θεωρήματος Κυριαρχίας έπεται ότι: $T(n)=\Theta(n^3)$

Σύγκριση:

$f(n)$? $n^{\log_b a}$

$<$ → Α'ΘΚ

$=$ → Β'ΘΚ

$>$ → Γ'ΘΚ

Η **μέθοδος της επανάληψης** χρησιμοποιείται για την επίλυση της αναδρομικής σχέσης

$$\mathbf{T}(\boldsymbol{n}) = \begin{cases} \boldsymbol{\alpha T}\left(\frac{\boldsymbol{n}}{\boldsymbol{b}}\right) + \boldsymbol{f}(\boldsymbol{n}), & \boldsymbol{n} > \boldsymbol{n_0} \\ \boldsymbol{c}, & \boldsymbol{n} = \boldsymbol{n_0} \end{cases}$$

- Όταν το ζητάει ρητά η εκφώνηση
- Όταν αποτυγχάνει το θεώρημα κυριαρχίας
- Όταν δεν θέλουμε απλά μία ασυμπτωτική εκτίμηση

1. Κάνουμε 3 εφαρμογές της αναδρομικής σχέσης (μέχρι να φτάσουμε στη μορφή: $\mathrm{T(n)} = a^3 T\left(\frac{n}{b^3}\right)$). Χρήσιμο το πρόχειρο

2. Εκτίμηση της σειράς που προκύπτει μετά από k επαναλήψεις (μας καθοδηγεί ο όρος: $T\left(\frac{n}{b^k}\right)$)

3. Υπολογίζουμε πότε σταματάει η αναδρομή (Θέτουμε $\frac{n}{b^k} = n_0$ και λύνουμε ως προς k). Π.χ. αν $n_0 = 1$, τότε $\mathrm{k} = \log_b n$

4. Αντικατάσταση του k στον τύπο του βήματος 2

5. Υπολογισμός της σειράς που προκύπτει. Χρήσιμος ο τύπος: $\sum_{i=0}^{n} x^i = \frac{x^{n+1}-1}{x-1}$

Πρόχειρο:

$$\mathrm{T}(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3T\left(\frac{n}{4^2}\right) + \left(\frac{n}{4}\right)^2$$

$$T\left(\frac{n}{4^2}\right) = 3T\left(\frac{n}{4^3}\right) + \left(\frac{n}{4^2}\right)^2$$

<u>Παράδειγμα:</u> Να λύσετε την αναδρομή:

$$\mathrm{T}(n) = \begin{cases} 3T\left(\frac{n}{4}\right) + n^2 & n > 1 \\ 0 & n = 1 \end{cases}$$

Λύση:

$$\mathrm{T}(n) = 3T\left(\frac{n}{4}\right) + n^2 = 3\left[3T\left(\frac{n}{4^2}\right) + \left(\frac{n}{4}\right)^2\right] + n^2$$

$$= 3^2 T\left(\frac{n}{4^2}\right) + 3\left(\frac{n}{4}\right)^2 + n^2 = 3^2\left[3T\left(\frac{n}{4^3}\right) + \left(\frac{n}{4^2}\right)^2\right] + 3\left(\frac{n}{4}\right)^2 + n^2$$

$$= 3^3 T\left(\frac{n}{4^3}\right) + 3^2\left(\frac{n}{4^2}\right)^2 + 3\left(\frac{n}{4}\right)^2 + n^2 =$$

$$= \cdots =$$

$$= 3^k T\left(\frac{n}{4^k}\right) + 3^{k-1}\left(\frac{n}{4^{k-1}}\right)^2 + \cdots + 3^2\left(\frac{n}{4^2}\right)^2 + 3\left(\frac{n}{4}\right)^2 + n^2 =$$

Η αναδρομή σταματά όταν $\frac{n}{4^k} = 1 \Rightarrow \mathrm{n} = 4^k \Rightarrow \mathrm{k} = \log_4 n$

$$= 3^{\log_4 n} T\left(\frac{n}{4^{\log_4 n}}\right) + 3^{\log_4 n - 1}\left(\frac{n}{4^{\log_4 n - 1}}\right)^2 + \cdots + 3^2\left(\frac{n}{4^2}\right)^2 + 3\left(\frac{n}{4}\right)^2 + n^2 =$$

$$= 3^{\log_4 n} T(1) + 3^{\log_4 n - 1}\left(\frac{n}{4^{\log_4 n - 1}}\right)^2 + \cdots + 3^2\left(\frac{n}{4^2}\right)^2 + 3\left(\frac{n}{4}\right)^2 + n^2 =$$

$$= \sum_{i=0}^{\log_4 n - 1} 3^i \left(\frac{n}{4^i}\right)^2 = \sum_{i=0}^{\log_4 n - 1} 3^i \frac{n^2}{(4^i)^2} = n^2 \sum_{i=0}^{\log_4 n - 1} \frac{3^i}{(4^2)^i} =$$

$$= n^2 \sum_{i=0}^{\log_4 n - 1} \frac{3^i}{16^i} = n^2 \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i =$$

$$= n^2 \frac{\overbrace{\left(\frac{3}{16}\right)^{\log_4 n}}^{0} - 1}{\frac{3}{16} - 1} = \Theta(n^2)$$