Διαίρει και Βασίλευε

```
function Solution (I);
begin
    if size(I) ≤ smallSize
        then Solution := DirectSolution(I)
        else
            Decompose(I, I_1, . . . , I_k);
            for i := 1 to k do
                S_i := Solution (I_i)
            end { for };
            Solution := Combine(S_1, . . . . S_k)
    end { if }
end { Solution }
```

$$T(n) = \begin{cases} S(n) & \text{for } n \le smallSize \\ D(n) + \sum_{i=1}^{k} T(size(I_i)) + C(n) & \text{for } n > smallSize \end{cases}$$

T(n): το κόστος του αλγόριθμου

S(n): το κόστος της απευθείας λύσης

D(n): το κόστος της διαίρεσης του αρχικού προβλήματος σε μικρότερα προβλήματα

C(n): κόστος συνδυασμού των λύσεων

- Συνήθως ισοδιάσπαση σε δύο προβλήματα ίσου μεγέθους

$$T(n) = \begin{cases} S(n) & \text{for } n \le smallSize \\ D(n) + 2T\left(\frac{n}{2}\right) + C(n) & \text{for } n > smallSize \end{cases}$$

Πολλές φορές η διάσπαση αυτή οδηγεί στην βέλτιστη λύση. Π.χ. όταν T και T' αυστηρά αύξουσες συναρτήσεις: $T(m)=cm^r$, $T(m)=mln(m)$

$n_1+n_2=n$, $T(n_1)+T(n_2)= T(n_1)+T(n-n_1)=f(n_1)$

Για ελάχιστο της f: $0 = f'(n_1) = T'(n_1) - T'(n-n_1)$, δηλαδή

$$T'(n_1) = T'(n-n_1).$$

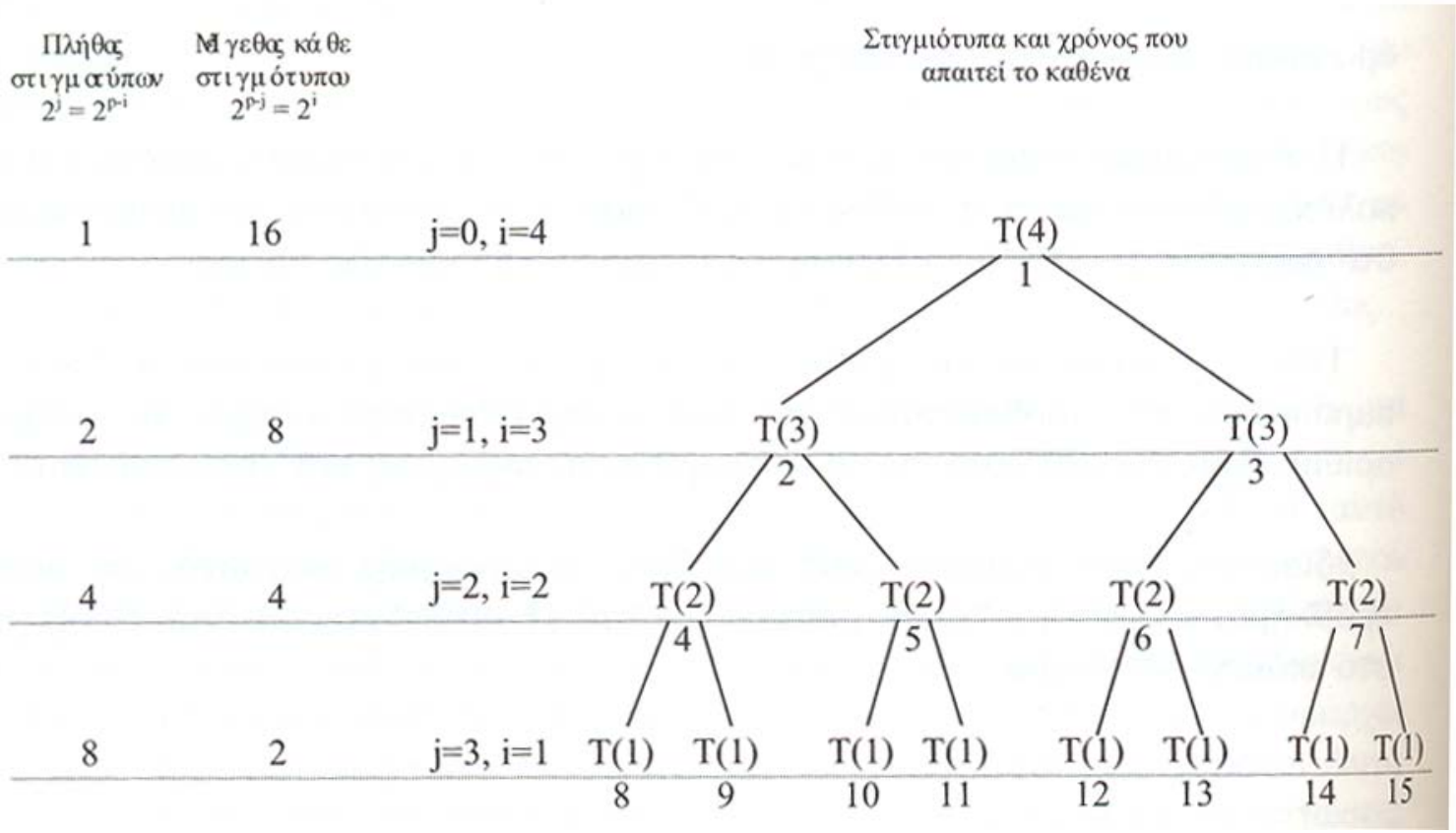
T' είναι αυστηρά αύξουσα, $n_1 = n - n_1$ δηλαδή $n_1 = n/2 = n_2$.

επειδή T' αύξουσα, ισχύει: $f''(n_1) = T''(n_1) + T''(n - n_1) > 0$.

άρα πρόκειται πράγματι για ελάχιστο.

Γενική ανάλυση της μεθόδου Δ & Β

$n = 2^p, \quad p = \log n.$



Λύση ενός υποπροβλήματος:

Λύση ενός υποπροβλήματος $T(i) = T(i-1) + E(i)$, όπου $i = i_o+1,\ldots,p$ $T(i_o) = b$,

Η λύση της αναδρομικότητας (2.2-1), με την αρχική συνθήκη (2.2-2), γίνεται εύκολα με την **αλλεπάλληλη εφαρμογή** της:

$$T(i) = T(i-1)+E(i) = [T(i-2)+E(i-1)]+E(i) = T(i-2)+E(i-1)+E(i) =$$
$$[T(i-3)+E(i-2)]+E(i-1)+E(i) = \ldots,$$

οπότε τελικά προκύπτει η λύση $T(i)= b+ \sum_{j=i_o+1}^{i} E(j)$ $i = i_o,\ldots,p,$

$$T(p)= b+ \sum_{j=i_o+1}^{p} E(j).$$

Λύση και των δύο υποπροβλημάτων: $T(i) = 2T(i-1) + E(i).$

Αυτό όμως πρέπει να γίνει για όλα τα στιγμιότυπα της διάσπασης i, που είναι σε πλήθος 2^{p-i}. Αν λοιπόν καλέσουμε $\tau(i)$ τον χρόνο που απαιτείται συνολικά, για όλα τα στιγμιότυπα της διάσπασης i, έχουμε

$$\tau(i)= 2^{p-i}\,T(i)= 2^{p-i}\{2T(i-1)+E(i)\}= 2^{p-i+1}\,T(i-1)+2^{p-i}\,E(i).$$ Επειδή

$$\tau(i-1) = 2^{p-(i-1)}\,T(i-1), \qquad \tau(i) = \tau(i-1) + 2^{p-i}\,E(i), \quad i=i_o+1,\ldots,p.$$

Η αρχική συνθήκη είναι

$$\tau(i_o) = 2^{p-i_o} T(i_o) = 2^{p-i_o} b$$

$$\tau(i) = 2^{p-i_o} b + \sum_{j=i_o+1}^{i} 2^{p-j} E(j),$$

$$\tau(p) = 2^{p-p} T(p) = T(p) \Rightarrow T(p) = \tau(p) = 2^{p-i_o} b + \sum_{j=i_o+1}^{p} 2^{p-j} E(j).$$

Αλγόριθμος του Strassen για πολλαπλασιασμό πινάκων

Ας θεωρήσουμε πίνακες $n \times n$. Η παρακάτω μέθοδος βρίσκει το γινόμενο $A \cdot B = C$ για πίνακες 2×2 με 7 πολλαπλασιασμούς, αντί για 8 που συνήθως απαιτούνται:

Πρώτα βρίσκουμε τις ποσότητες

$$x_1 = (a_{11} + a_{22})(b_{11} + b_{22}) \qquad x_5 = (a_{11} + a_{12})b_{22}$$
$$x_2 = (a_{21} + a_{22})b_{11} \qquad x_6 = (a_{21} - a_{11})(b_{11} + b_{12})$$
$$x_3 = a_{11}(b_{12} - b_{22}) \qquad x_7 = (a_{12} - a_{22})(b_{21} + b_{22})$$
$$x_4 = a_{22}(b_{21} - b_{11})$$

και μετά υπολογίζουμε τα στοιχεία c_{ij} του C από τις σχέσεις:

$$c_{11} = x_1 + x_4 - x_5 + x_7 \qquad c_{12} = x_3 + x_5,$$
$$c_{21} = x_2 + x_4 \qquad c_{22} = x_1 + x_3 - x_2 + x_6$$

Όταν $n = 2^k$ (δύναμη του 2) τότε μια ισοδιάσπαση των πινάκων σε υποπίνακες,

$$A \cdot B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = C,$$

τώρα τα a_{ij}, b_{ij}, x_i, c_{ij} είναι πίνακες $2^{k-1} \times 2^{k-1}$.

$$\mu(k) = 7 \cdot \mu(\kappa - 1), \quad \mu > 0$$

$$\mu(0) = 1,$$

για τους πολλαπλασιασμούς, ενώ για τις προσθαφαιρέσεις ισχύει

$$a(k) = 18(2^{k-1})^2 + 7a(k-1), \quad k > 0$$

$$a(0) = 0.$$

Η λύση των αναδρομικών σχέσεων δίνει:

$$\mu(k) = 7^k = 7^{\log n} = n^{\log 7} \approx n^{2.81} < n^3$$

$$a(k) = 18(2^{k-1})^2 + 7a(k-1) = 18(2^{k-1})^2 + 7 \cdot 18(2^{k-2})^2 + 7^2 a(k-2)$$

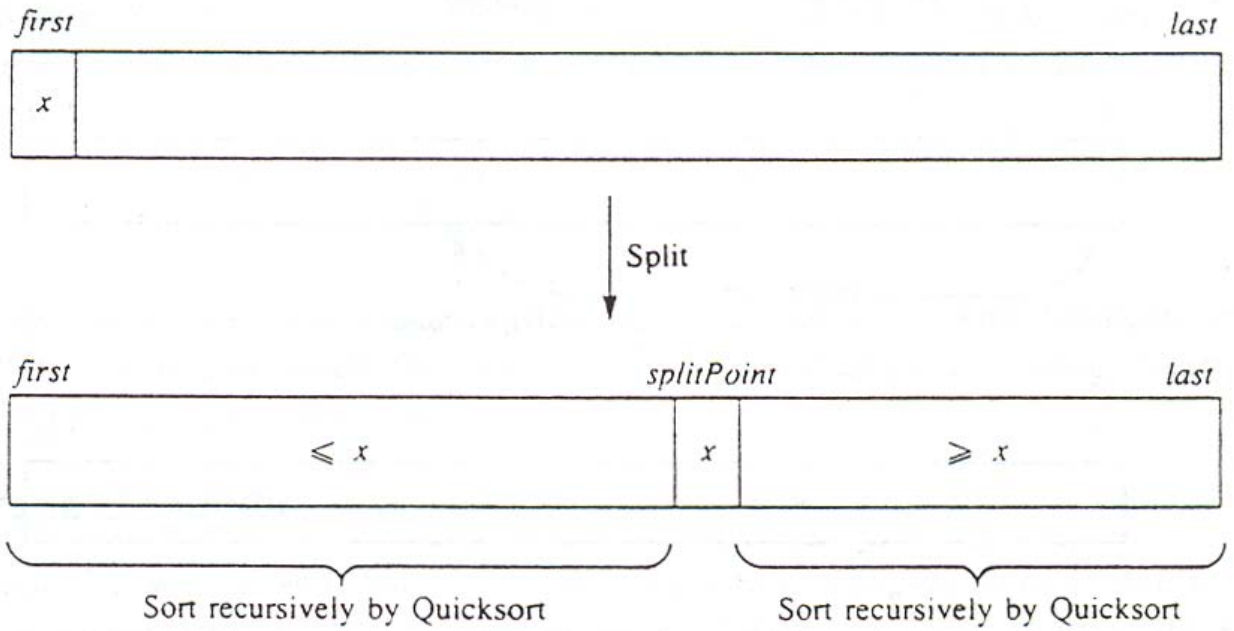
$$= 18(2^{k-1})^2 + 7 \cdot 18(2^{k-2})^2 + 7^2 \cdot 18(2^{k-3})^2 + 7^3 a(k-3) = \ldots$$

$$= \sum_{i=0}^{k-1} 7^i \cdot 18(2^{k-i-1})^2 = 18(2^k)^2 \sum_{i=0}^{k-1} \frac{7^i}{(2^{i+1})^2} =$$

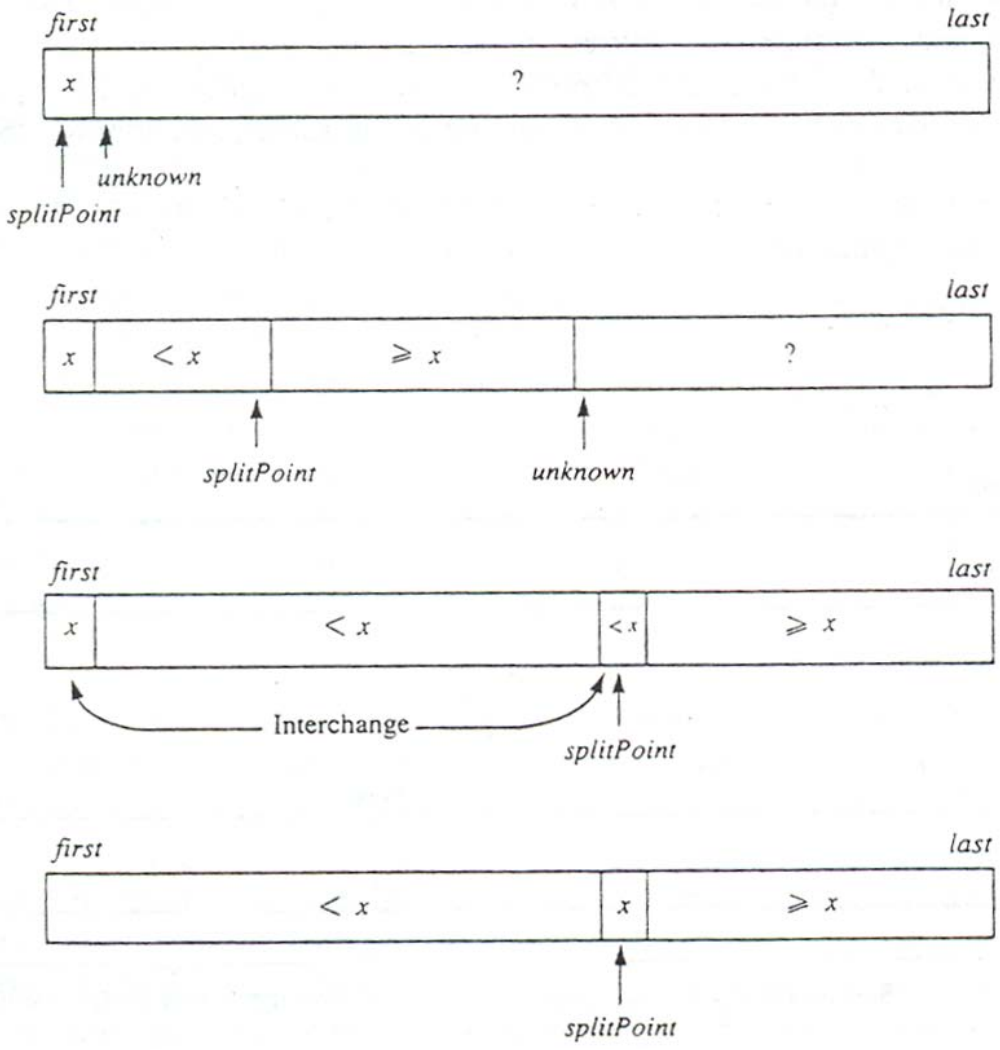
$$= (q/2)(2^k)^2 \sum_{i=0}^{k-1} (7/4)^i = (q/2)(2^k)^2 \frac{(7/4)^k - 1}{(7/4) - 1} =$$

$$= 6 \cdot 7^k - 6 \cdot 4^k \qquad a(k) \approx 6n^{2.81} - 6n^2 < n^3$$

Δ & Β για διάταξη λίστας. Αλγόριθμος Quicksort




```
procedure Quicksort (first, last : Index);
var
    splitPoint: Index;
    begin
        if first < last then
            Split (first, last, splitPoint);
            Quicksort (first, splitPoint-1);
            Quicksort (splitPoint+1, last)
        end { if }
    end { Quicksort }
```

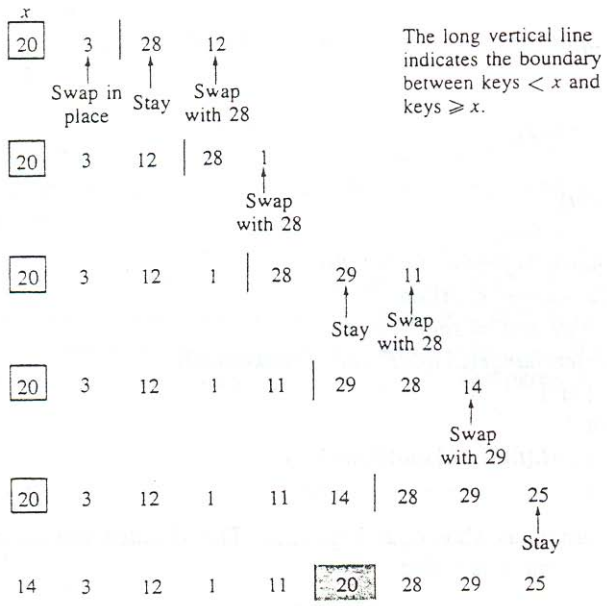

first
last
x
?
unknown
splitPoint
first
last
x
< x
≥ x
?
splitPoint
unknown
first
last
x
< x
< x
≥ x
Interchange
splitPoint
first
last
< x
x
≥ x
splitPoint

```
procedure Split (first, last: Index; var splitPoint: Index);
var
    x: Key;
    unknown: Index;
begin
    x := L[first];
    splitPoint := first;
    for unknown := first+1 to last do
        if L[unknown] < x then
            splitPoint := splitPoint+1;
            Interchange(L[splitPoint], L[unknown])
        end { if }
    end { for };
    Interchange(L[first], L[splitPoint])
end { Split }
```

The keys

20 3 28 12 1 29 11 14 25

The first execution of *Split*



Split the first section (details not shown).

11 3 12 1 14

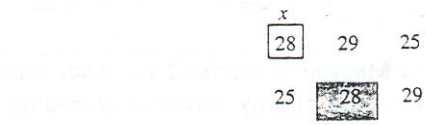
Split the first section.

1 3 11 12

Split the first section.

1 3

Sections with only one key are sorted. Now return to the second section from the first split.

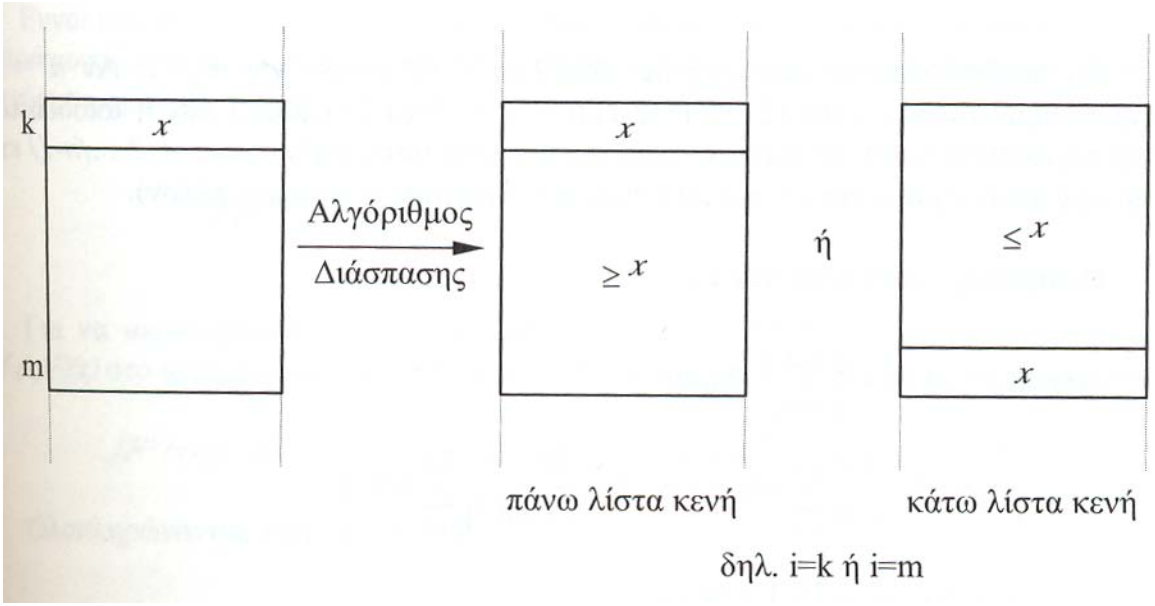


Sections with one key are sorted.

Final list

1 3 11 12 14 20 25 28 29

ΠΧΠ



$$1+2+\ldots+(n-1)=(n-1)n/2=\theta\left(n^2/2\right)$$ συγκρίσεις

ΠΜΟ

η μέση τιμή για όλες τις (ισοπίθανες) περιπτώσεις ως προς i είναι

$$\frac{1}{n}\sum_{i=1}^{n}\left[\Pi MO(i-1)+\Pi MO(n-i)\right]$$

Προσθέτοντας και τις αρχικές $n-1$ συγκρίσεις βρίσκουμε την αναδρομική ανισότητα

$$\Pi MO(n) \le n-1+\frac{2}{n}\sum_{i=0}^{n-1}\Pi MO(i), \qquad n \ge 2,$$

$$\Pi MO(0)=0, \quad \Pi MO(1)=0.$$

Θα αποδείξουμε με επαγωγή ότι $\Pi MO(n) \le 3/2\ n\log n$, για $n \ge 2$.

Αν $n = 2$, $\Pi MO(2)=2-1+(2/2)\cdot 0 = 1 < 3=(3/2)\ (2\log 2)$

για $n \ge 3$

$$\Pi MO(n) \le n-1+\frac{2}{n}\sum_{i=2}^{n-1}\frac{3}{2}i\log i$$

$$= n-1+\frac{3}{2}\frac{2}{n}\sum_{i=2}^{n-1} i\log i = n-1+\frac{3}{2n}\frac{2}{\ln(2)}\sum_{i=2}^{n-1} i\ln(i)$$

$$\le n-1+\frac{3}{2n}\frac{2}{\ln(2)}\int_2^n x\ln(x)\,dx$$

$$\int_2^n x\ln(x)\,dx=\left[\frac{1}{2}x^2\ln(x)-\frac{x^2}{4}\right]_2^n=\frac{1}{2}n^2\ln(n)-\frac{n^2}{4}-1$$

$$\Pi MO(n) \le \frac{3}{2}n\log n$$