$\frac{1}{2n} T(n) = O(\lg^2 n)$

$\Rightarrow T(n) = O(2n \cdot \lg^2 n) = O(n \lg^2 n)$ ✓

12 Μαΐου 2023

Διαίρει & ~~Βασίλευε~~ Κυρίευε (Βασίλευε)

Divide & Conquer

Παράδειγμα

```
factorial(n)
  if n=0 then
     return 1
  else
     return n*factorial(n-1);
  end-if
end-function
```

← Αν ξέρω λύση αυτού...

Μπορώ να υπολογίσω αυτό/να τη λύση

• Τυποποίηση αναδρομικών σχέσεων

Διαίρει: Διαιρείται το πρόβλημα σε μικρ. στιγμ/να *

Κυρίευε: Τα μικρά στιγμιότυπα αυτά επιλύονται αναδρομικά για πολύ ~~Συνδύασε~~ μικρά έχουμε τετριμμένες λύσεις

Συνδύασε: Συνδυασμός λύσεων στιγμιοτύπων μικρ. με τρόπο τέτοιο ώστε να προκύπτει λύση του παρόντος στιγμιοτύπου

* μικρότερα της τάξης μεγέθους του προβλήματος

• Να επικεντρωθούμε στη γενικότερη μεθοδολογία και όχι συγκεκριμένα παραδείγματα (γιατί θα βγάζει πιθανότατα δουλειά αυτής της μεθοδολογίας που να μην έχουμε κάνει)

$\boxed{\text{Δυαδική Αναζήτηση}}$

• n στοιχεία ταξινομημένα, ψάχνω a που μπορεί να βρίσκεται αριστερά ή δεξιά από την εκάστοτε θέση που επιλέγεται

π.χ. για $a=2$

$\underbrace{1\ 2\ 3}_{\uparrow\ \checkmark} \leftarrow \underset{\uparrow\ \times}{4} \rightarrow \underbrace{5\ 6\ 7}_{\uparrow\ \times}$

· Η εκάστοτε αυτή θέση συνηθίζεται να είναι η μέση της εκάστοτε σειράς αριθμών, όπως στο παράδειγμα

Απαιτούνται: A, lo, hi, για lo ≤ hi, και α

Επιστρέφεται: Θέση του ζητούμενου στοιχείου, αλλιώς κάποια τυχαία τιμή (π.χ. -1)

(A, lo, hi (lo ≤ hi), α)

```
function BSearch(int α, int A[], int lo, int hi)
{
  if (lo = hi) then        τετριμμένο (trivial) σημείο
     if A[lo] = a then
        return lo;
     else return 0;
  else
     mid ← ⌊(lo+hi)/2⌋;
     if a ≤ A[mid] then
        return BSearch(a, A, lo, mid);
     else
        return BSearch(a, A, mid+1, hi);
     end_if
  end_if
end_function
```

Δε συνδυάζεται καμία λύση, γιατί λύνουμε είτε το ένα είτε το άλλο BSearch

· Άρα, ζητάω να λύσει ένα εκ των δύο αυτών σημείων

$n = hi - lo + 1;$

$mid \leftarrow \lceil \frac{n}{2} \rceil$

$m \leftarrow \lceil \frac{hi+lo}{2} \rceil$

ΑΣΚΗΣΗ: Να δείξετε τι σχέση έχει το $\lfloor \frac{lo+hi}{2} \rfloor$ με το $n/2$, και γιατί έχει σχέση με ~~αυτό~~

$\boxed{\text{Ταξινόμηση}}$

$lo \le k \le hi$

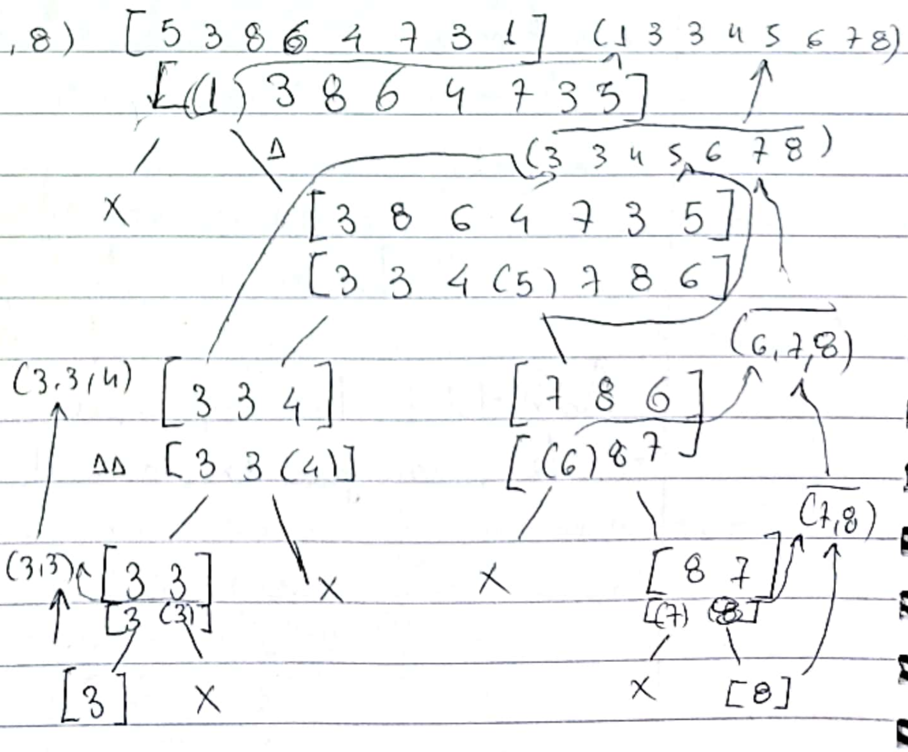
Recap: $j \leftarrow$ Partition(A, lo, hi, k)

- Δουλεύει ώστε το στοιχείο που θα ~~εισαχθεί στη~~ μετακινηθεί στη σειρά να είναι ένα μεγαλύτερο από κάθε στοιχείο αριστερά του και μικρότερο από κάθε στοιχείο δεξιά του
- Αν πάω με Διαίρει και Κυρίευε, θα δείξω τις λύσεις και των δύο στιγμιοτύπων για να έχω το συνδυασμό των λύσεων αυτών

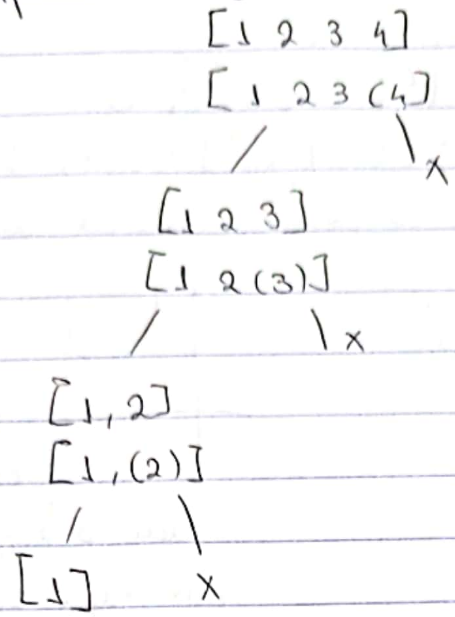
```
QuickSort(int A , int lo, int hi)
    if lo<hi then
      k<-hi;
      j<-partition(A, lo, hi, k);
      Quicksort(A, lo, j-1);
      Quicksort(A, j+1, hi);
    end_if
end_function
```

π.χ. A = [5 3 8 6 4 7 3 1]

Quicksort(A, 1, 8)



· Αν η σειρά ήταν ταξινομημένη:

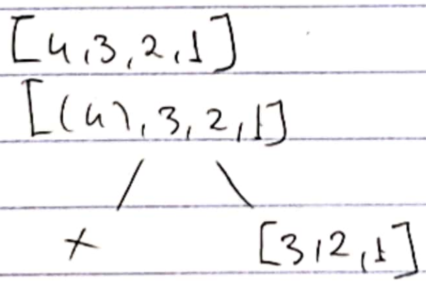


Αν T(n) αριθμός ΣΥΒ, τότε:

$$T(n)=\begin{cases} T(n-1)+\Theta(n) & , \ n\geq 2\\ 1 & , \ n\leq 1\end{cases}$$

$$T(n) = T(n-1)+cn = T(n-2)+c\cdot n-1+c\cdot n = \cancel{T(n-k}$$
$$= T(n-k) + \sum_{j=0}^{k-1}(n-j-1) = \Theta(n^2)$$

Έχω $\Theta(n^2)$ και όταν είναι αντίστροφα ταξινομημένη η σειρά



$$T(n) = \max\{T(q)+T(n-q+1)\} + \Theta(n)$$

• Υποθέτω ότι: $T(n) = O(n^2)$

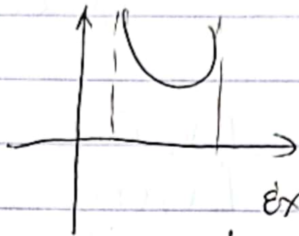
$T(q) \leq cq^2$ (δηλ. $T(q) \leq cq^2$)

$T(n-q-1) \leq c \cdot (n-q-1)^2$

$$T(n) \leq \max\{c \cdot q^2 + c(n-q-1)^2\} + \Theta(n)$$
$$= c\max\{q^2 + (n-q-1)^2\} + \Theta(n) \leq c \cdot n^2$$

• Εδώ, έστω το q το μέγεθος ενός εκ των δύο στιγμιοτύπων

$f(q) = q^2 + (n-q-1)^2$



• Επειδή είναι δευτεροβάθμια, έχουμε καμπύλη κυρτή και το μέγιστο το επιτυγχάνει στα άκρα του πεδίου ορισμού της

• Αν είναι για πλήρες δέντρο (κάθε σωστή θέση τυχαίνει ως διάμεσος):

$$T(n) = \begin{cases} T(\lfloor \frac{n-1}{2} \rfloor) + T(\lceil \frac{n-1}{2} \rceil) + \Theta(n), & n \geq 2 \\ 1, & n \leq 1 \end{cases}$$

$\Theta(n \lg n)$

$\Omega(n \lg n) = T(n) = O(n^2)$