ΠΑΡΑΔΕΙΓΜΑ Π1.10-1. Υπολογισμός Πολυπλοκότητας για Μερικά Αθροίσματα.

Ισχύει για τον πρώτο αλγόριθμο

$$ΠΜΟ(n)=ΠΧΠ(n) = n(n+1)/2 = O(n^2) = \theta(n^2/2)$$

και για τον δεύτερο

$$ΠΜΟ(n) = ΠΧΠ(n) = n\text{-}1 = O(n) = \theta(n),$$

ΠΑΡΑΔΕΙΓΜΑ Π1.10-2. Πολυπλοκότητες για Διαδοχική Αναζήτηση.

$$ΠΧΠ(n) = n \text{ συγκρίσεις}$$

Για να βρούμε την *ΠΜΟ* πρέπει να θεωρήσουμε **όλες τις δυνατές περιπτώσεις τερματισμού και την πιθανότητα που έχει για να συμβεί η κάθε μία.**

n=2, Όλες οι δυνατές περιπτώσεις είναι τρεις:

($Π_1$) Να είναι $x=L(1)$, οπότε ο αλγόριθμος τερματίζει μετά από $t(Π_1)=1$ σύγκριση,

($Π_2$) Να είναι $x=L(2)$, οπότε ο αλγόριθμος τερματίζει μετά από $t(Π_2)=2$ συγκρίσεις, και

($Π_3$) Το x να μην είναι στην λίστα, οπότε πάλι ο αλγόριθμος τερματίζει μετά από $t(Π_3)=2$ συγκρίσεις.

$$p(Π_1) = p(Π_2) = p(Π_3) = 1/3$$

$$\begin{aligned} ΠΜΟ(n=2) &= [t(Π_1) + t(Π_2) + t(Π_3)]/3 \\ &= 5/3 \\ &= p(Π_1)\, t(Π_1) + p(Π_2)\, t(Π_2) + p(Π_3)\, t(Π_3) \end{aligned}$$

Γενικά: $$ΠΜΟ(n) = \sum p(\Pi_i) \cdot t(\Pi_i).$$

$$\sum p(\Pi_i) = 1$$

Ας υπολογίσουμε τώρα την *ΠΜΟ* όταν η πιθανότητα να είναι το x στην λίστα είναι q και όλες οι περιπτώσεις $x=L(i)$, $i=1,\ldots,n$ είναι ισοπίθανες. Παρατηρούμε ότι υπάρχουν $n+1$ περιπτώσεις:

Περιπτώσεις Π_i, $i = 1,\ldots,n$ να είναι $x=L(i)$, με πιθανότητα $p(\Pi_i)=q/n$,
Περίπτωση Π_{n+1} να μην είναι το x στην λίστα, με πιθανότητα $1-q$.

$$\sum_{i=1}^{n} p(\Pi_i) + 1 - q = \sum_{i=1}^{n} (q/n) + 1 - q = 1$$

$$t(\Pi_i) = i, \ \ \text{για } i=1,\ldots,n \ \ \text{και} \ \ t(\Pi_{n+1}) = n,$$

$$ΠΜΟ(n) = \sum_{i=1}^{n} (q/n) i + (1-q)n = (q/n) \sum_{i=1}^{n} i + (1-q)n,$$

δηλαδή $ΠΜΟ(n) = q(n+1)/2 + (1-q)n$ συγκρίσεις.

Παράδειγμα - Βελτιωμένος Αλγόριθμος Διαδοχικής Αναζήτησης

- Τα στοιχεία της λίστας διατεταγμένα
- Αν σε μία σύγκριση x < L(i), ο αλγόριθμος τερματίζει αφού:
    - x < L(k) για k>i

Περίπτωση Π_1: $t(\Pi_1)=1$ σύγκριση, αν $x \leq L(1)$.

Περίπτωση Π_i: $t(\Pi_i)=i$ συγκρίσεις, αν $L(i-1) < x \leq L(i)$.

Περίπτωση Π_n: $t(\Pi_n)=n$ συγκρίσεις, αν $L(n-1) < x$

$$p(\Pi_i)=1/n$$

$$\Pi MO(n)=\sum_{i=1}^{n} i/n = 1/n \cdot n(n+1)/2 = (n+1)/2$$

Ας δεχθούμε τώρα ότι η πιθανότητα να είναι το x στην λίστα είναι q και ότι όλες οι περιπτώσεις επιτυχίας $E(i)$ είναι μεταξύ τους ισοπίθανες και, ακόμα, ότι όλες οι περιπτώσεις αποτυχίας $A(i)$, $i=1,\ldots,(n+1)$ είναι μεταξύ τους ισοπίθανες. Όλες οι $A(i)$ έχουν συνολική πιθανότητα $1-q$. Άρα:

$$p(A(i)) = (1-q)/(n+1),$$

$$p(E(i)) = q/n.$$

$$\Pi MO(n) = \sum_{i=1}^{n+1} p(A(i))t(A(i)) + \sum_{i=1}^{n} p(E(i))t(E(i))$$

$$= \frac{1-q}{n+1}\sum_{i=1}^{n+1} t(A(i)) + \frac{q}{n}\sum_{i=1}^{n} t(E(i)) = \frac{1-q}{n+1}(1+\ldots+n+n) + \frac{q}{n}(1+\ldots+n)$$

$$= \frac{1-q}{n+1}\left(n+\frac{n(n+1)}{2}\right) + \frac{q}{n}\cdot\frac{n(n+1)}{2} = \frac{(1-q)\cdot n}{n+1} + \frac{n}{2} - \frac{q\cdot n}{2} + \frac{q\cdot n}{2} + \frac{q}{2},$$

$$\Pi MO(n) = \frac{n}{2} + \frac{(1-q)\cdot n}{n+1} + \frac{q}{2} = \theta\left(\frac{n}{2}\right) \text{ συγκρίσεις.}$$

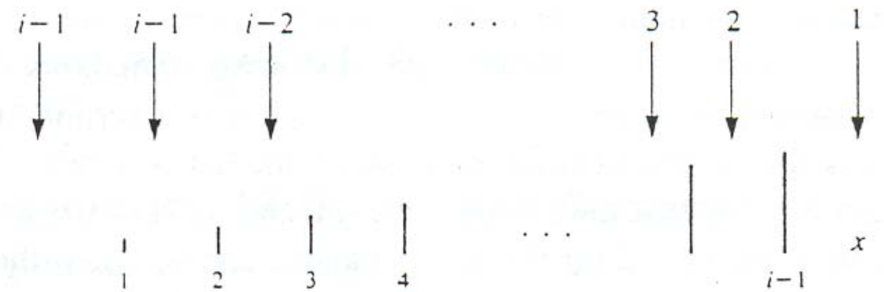
ΠΑΡΑΔΕΙΓΜΑ Π1.10-8. Πολυπλοκότητες της μεθόδου παρεμβολής για διάταξη λίστας.

$$ΠΧΠ(n) = \sum_{i=2}^{n}(i-1) = n(n-1)/2 = \Theta(n^2).$$

Έστω τώρα ότι τα *i-1* έχουν ήδη διαταχθεί και πρέπει να γίνει παρεμβολή για το νέο στοιχείο $x = x_i$. Η πιθανότητα να χρειασθούν *k* συγκρίσεις, *k=1,...,(i-1)*, για την εύρεση της σωστής θέσης για το x_i είναι *1/i*.

Πράγματι. Η πιθανότητα αυτή είναι ίση με $a/n! = 1/i$. όπου

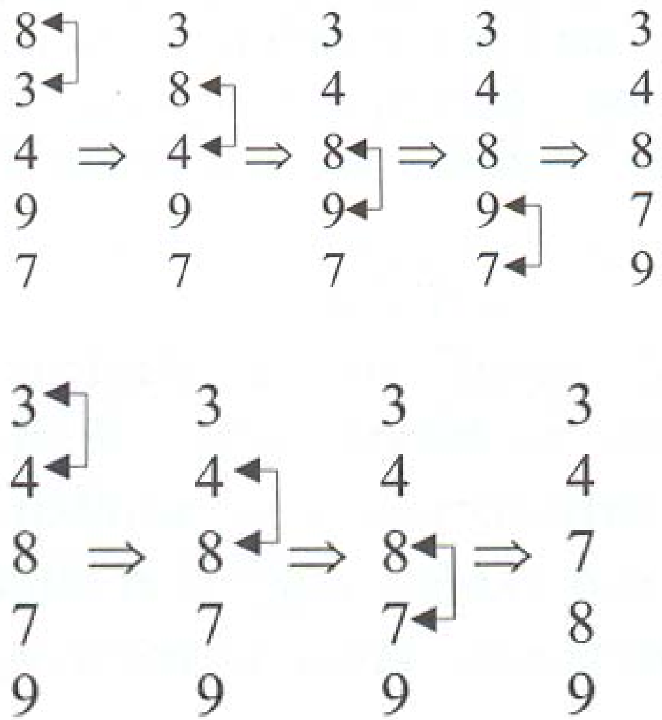
$$a = \binom{n}{i}(i-1)!(n-i)! = \frac{n!}{i},$$



$$\sigma_i = \Sigma\,(\text{πιθανότητα περίπτωσης}) * (\text{αριθμός συγκρίσεων περίπτωσης})$$
$$= \sum_{j=1}^{i-2}\frac{1}{i}j + \frac{2}{i}\,(i-1) = (i-1)\,(i+2)\,/\,2i.$$

$$ΠΜΟ(n) = \sum_{i=2}^{n}\sigma_i = \frac{1}{2}\sum_{i=2}^{n}(i+1) - \sum_{i=2}^{n}\frac{1}{i} = \frac{n^2+3n}{4} - \sum_{i=2}^{n}\frac{1}{i}.$$

ΠΑΡΑΔΕΙΓΜΑ Π1.10-9. Bubblesort.



Αλγόριθμος Bubblesort

ΕΙΣΟΔΟΣ: *n≥1* και διάνυσμα *L* από *n* αντιπροσώπους

ΕΞΟΔΟΣ: διάνυσμα *L* διαταγμένο σε μη φθίνουσα σειρά

```
for k=n-1 down to 1 do
    T=0
    for j = 1 to k do
        if L(j) > L(j+1) then
            a=L(j)
            L(j)=L(j+1)
            L(j+1) = a
            T=1
        end -"if-then"
    end j-loop
    if T=0 exit k-loop
end k-loop
```

Ανάλυση

ΠΜΟ(n)≥(n-1)n/4=O(n^2)

Ε: ο αριθμός των εναλλαγών που υλοποιεί ο αλγόριθμος

χρόνος εκτέλεσης του Bubble Sort ≥ E

$$E = \sum_{i=1}^{n} \varepsilon(i)$$

όπου ε(i): το πλήθος των L(j) (j>x) που είναι μικρότερα του i και L(x)=i

Ισχύει προφανώς $\varepsilon(1) = 0$ και γενικότερα $0 \le \varepsilon(i) \le (i-1)$ για $i=1,\dots,n$. Υπάρχουν συνολικά $n!$ διανύσματα $\varepsilon = [\varepsilon(1),\dots,\varepsilon(n)]^T$ με τις ιδιότητες αυτές, όσα και οι μεταθέσεις των n στοιχείων. Μάλιστα, τα διανύσματα ε βρίσκονται σε αντιστοιχία "ένα προς ένα" με τις μεταθέσεις/στιγμιότυπα του προβλήματος.

$X(n) := \frac{1}{n!}$ {άθροισμα όλων των E} $= \frac{1}{n!}$ {άθροισμα των διαφορετικών μόνο E επί την πολλαπλότητα εμφάνισης τους $F(E)$} $= \frac{1}{n!}\sum_{E=0}^{S_n} E(n)F(E(n))$,

όπου S_n είναι η μέγιστη τιμή του E. Εύκολα βρίσκουμε ότι $S_n=(n-1)n/2$.

Ανάλυση

$$X(n)=\frac{1}{n!}\sum_{E'=0}^{Sn}E'F(E')=\frac{1}{n!}\{[0+\ldots+(n-1)]F(0)+[1+\ldots+n]F(1)+[2+\ldots+(n+1)]F(2)$$
$$+\ldots+[S_{n-1}+\ldots+(S_{n-1}+n-1)]F(S_{n-1})\}.$$

Αφού όμως $\kappa+\ldots+[\kappa+(n+1)]=n\kappa+[0+\ldots+(n-1)]=n\kappa+n(n-1)/2=n(2\kappa+n-1)/2$,

βρίσκουμε

$$X(n)=\frac{n}{2n!}\{(0+n-1)F(0)+(2+n-1)F(1)+(4+n-1)F(2)+\ldots+(2S_{n-1}+n-1)F(S_{n-1})\}=$$
$$\frac{1}{(n-1)!}\{0F(0)+1F(1)+\ldots+S_{n-1}F(S_{n-1})\}+\frac{n}{2n!}(n-1)\{F(0)+\ldots+F(S_{n-1})\}$$

$$X(n)=X(n-1)+\frac{n}{2}\quad\Longrightarrow$$

$$X(n)=X(n-1)+\frac{n}{2}=\left[X(n-2)+\frac{n-1}{2}\right]+\frac{n}{2}$$
$$=X(n-3)+\frac{n-2}{2}+\frac{n-1}{2}+\frac{n}{2}=\cdots$$
$$=X(1)+\frac{1}{2}(2+\cdots+n)=0+\frac{n(n+1)}{4}-\frac{1}{2},$$

$$X(n)=\frac{n(n+1)}{4}-\frac{1}{2}.$$

ΠΑΡΑΔΕΙΓΜΑ Π1.10-10. Δυαδική αναζήτηση.

Αλγόριθμος Δυαδικής Αναζήτησης

ΕΙΣΟΔΟΣ: $n \geq 1$, array L μήκους n, [όπου $L(i)=x_i$ και $L(1) \leq \ldots \leq L(n)$] και x του ίδιου τύπου με τα $L(i)$, $i=1,\ldots, n$.

ΕΞΟΔΟΣ: $j \in \{1,\ldots,n\}$, αν υπάρχει $L(j)=x$, αλλιώς $j=0$.

```
k ← 1; m ← n
while  k ≤ m do
    j ← ⌊(k+m)/2⌋        (ο j είναι στη «μέση» μεταξύ k και m)
    if x = L(j) then return       (επιτυχία, επιστρέφει το j)
    if x < L(j) then m ← j-1      (διαλέγει το αριστερό «μισό»)
                else k ← j+1      (διαλέγει το δεξιό «μισό»)
end
j ← 0                             (αποτυχία)
```

(α) ΠΜΟ: Όπως και στο *Π1.10-3*, δεχόμαστε ότι οι n περιπτώσεις επιτυχίας έχουν συνολική πιθανότητα q και είναι μεταξύ τους ισοπίθανες:

Ανάλυση

$$p(E(i)) = q/n, \qquad i=1,\dots,n.$$

Δεχόμαστε και ότι οι $n+1$ πιθανότητες αποτυχίας είναι μεταξύ τους ισοπίθανες:

$$p(A(i)) = (1-q)/(n+1), \qquad i=1,\dots,n+1.$$

$$ΠΜΟ(n) = \sum_{i=1}^{n} p(E(i))t(E(i)) + \sum_{i=1}^{n+1} p(A(i))t(A(i)) \implies$$

$$ΠΜΟ(n) = q/n\sum_{i=1}^{n} t(E(i)) + \left[(1-q)/(n+1)\right]\sum_{i=1}^{n+1} t(A(i))$$

Έστω $\quad n = 2^r - 1,$

$$\sum_{i=1}^{n} t(E(i)) = \sum_{k=0}^{r-1}(k+1)2^k = \sum_{j=1}^{r} j2^{j-1} = 2\sum_{j=1}^{r} j2^{j-1} - \sum_{j=1}^{r} j2^{j-1} \implies$$

$$= \sum_{j=1}^{r} j2^j - \sum_{j=1}^{r} j2^{j-1} = 1\cdot 2 + 2\cdot 2^2 + 3\cdot 2^3 + \dots + (r-1)\cdot 2^{r-1} + r\cdot 2^r - 1\cdot 2^0 - 2\cdot 2^1 - 3\cdot 2^2 - \dots - r\cdot 2^{r-1}$$

$$= -1\cdot 2^0 - (2 + 2^2 + \dots + 2^{r-1}) + r\cdot 2^r = -1 - (2^r - 2)/(2-1) + r\cdot 2^r$$

$$= (r-1)\cdot 2^r + 1.$$

Ανάλυση

$$t(A(i))=r \quad \Longrightarrow \quad \sum_{i=1}^{n+1} t(A(i)) = (n+1)r.$$

$$ΠΜΟ(n) = q/n\{(r-1)2^r + 1\} + \left[(1-q)/(n+1)\right]\{(n+1)r\}.$$

$$r = \lfloor logn \rfloor + 1 \qquad ΠΜΟ(n) = \lfloor logn \rfloor + \lfloor logn \rfloor (q/n) + (1-q) + q/n$$

(β) ΠΧΠ:

$$ΠΧΠ(n) = 1 + \lfloor logn \rfloor \text{συγκρίσεις}$$