Δυναμικός Προγραμματισμός Ασκήσεις

Άσκηση) Να βρεθεί η υποσειρά με το μεγαλύτερο άθροισμα της παρακάτω σειράς.

$$A = [-9,\ 1,\ -5,\ 4,\ 3,\ -6,\ 7,\ 8,\ -2]$$

$S_j=-9$ $S_j=-8$ $S_j=-5$ $S_j=-1$ $S_j=2$ $S_j=-6$ $S_j=1$ $\underline{S_j=9}$ $S_j=-2$

$$\begin{array}{c} \ \ 1\ \ 2\ \ 3\ \ 4\ \ 5 \\ p = [1, 1, 3, 3, 4, \dots\ ] \end{array}$$

1) Ορισμός Υποπροβλημάτων

Διακρίνουμε τα υποπροβλήματα $S(:j)$ για $j = \{1, \dots, n\}$.
Το υποπρόβλημα $S(:j)$ για μια τιμή του j αναφέρεται στο πρόβλημα εύρεσης της σειράς μεγαλύτερου αθροίσματος η οποία τερματίζει στη θέση j του πίνακα $A[\,]$.

2) Καθορισμός επιλογών:

1) Είτε να προσθέσουμε ένα στοιχείο στην σειρά μέγιστου αθροίσματος
2) Είτε να ξεκινήσουμε μία νέα σειρά μέγιστου αθροίσματος

π.χ.: Εάν $A[j] = -3$ και $S_{j-1} = 1$, τότε $S_j = -2$

Εάν $A[j] = -1$ και $S_{j-1} = -3$, τότε $S_j = -1$

3) Ορισμός Αναδρομικής σχέσης:

$$S_j = \begin{cases} \max\{A[j],\ S_{j-1} + A[j]\}, & \text{για } j > 1 \\ A[1], & \text{για } j = 1 \end{cases}$$

4) Τοπολογική ταξινόμηση:

Κορυφές Γραφήματος: Τα υποπροβλήματα S_j
Ακμές Γραφήματος: (S_{j-1}, S_j) για $j = \{2, \dots, n\}$
Σειρά διάταξης κορυφών: Αύξουσα διάταξη, βάση των τιμών του δείκτη j

$$\begin{array}{c} S_1 = -9 \\ \downarrow \\ S_2 = -8 \\ \downarrow \\ S_3 = -5 \\ \downarrow \\ S_4 = -1 \\ \vdots \\ S_9 = -2 \end{array}$$

$$\begin{tikzcd} F_{j-2} \arrow[r] \arrow[rr, bend left] & F_{j-1} \arrow[r] & F_j \end{tikzcd}$$

(F_{j-1}, F_j), (F_{j-2}, F_j) για $j = \underbrace{\{2, \dots, n\}}_{\text{Σύνολο}}$

5) Επίλυση Αρχικού προβλήματος.

$$S = \max\{ S_j : j = [1, \dots, n\} \}$$

```
void    maxsumseries (int A[], int p[], int n) {
        int  S[n];
        S[1] ← A[1]
        for (j=2; j≤n; j++) do {

            if ( S[j-1] + A[j] > A[j] ) {
                   S[j] ← S[j-1] + A[j]
                   p[j] ← j-1
            }
            else {
                   S[j] ← A[j]
                   p[j] ← j
            }
        }
        int argmax ← 1;
        int max ← S[1]
        for (j=2; j≤n; j++ ) do {
            if ( max < S[j]) {
                   max ← S[j];
                   argmax ← j;
            }
        }

        int  curr  ← argmax
        while ( curr ≠ p[ curr ]) {
              print curr;
              curr ← p[curr ];
        }
        print curr;
```

```
                    curr
          ↓    ↓    ↓
curr 1  2  3    4    5  . . . .
p= [1, 1,  3,   3,   4, . . .      ]
```

Άσκηση) Να βρεθεί η μεγαλύτερη αύξουσα υπακολουθία της παρακάτω σειράς

Για τη σειρά

A	5	2	8	6	3	6	9	7
L	1	1	2	2	2	3	4	4

Υπακολουθία: Είναι η ακολουθία μιας αρχικής σειράς, όπου τα στοιχεία της δεν χρειάζεται να είναι διαδοχικά

Πχ: 2, 3, 6, 7

1) Ορισμός Υποπροβλημάτων: $L(i:j)$ το πρόβλημα εύρεσης της αύξουσας υπακολουθίας μέχρι το στοιχείο j.
Για $j=\{1,\dots,n\}$, $i=\{1,\dots,j-1\}$

2) Καθορισμός Επιλογών: $\underbrace{j-1}_{\text{Υποψήφιες υπακολουθίες}}+\underbrace{1}_{\text{Αρχικοποίηση νέας ακολουθίας}}=j$ Επιλογές

3) Αναδρομική σχέση:

$$L(i:j)=\begin{cases}\max\{L(i): i=\{1,\dots,j-1\}, A[i]<A[j]\} & \\ 1 & j=1\end{cases}$$

4) Τοπολογική ταξινόμηση: for $j=1$ to n

for $i=1$ to $j-1$

Αύξουσα διάταξη των υποπροβλημάτων, ακολουθώντας τις τιμές του j

5) Επίλυση αρχικού προβλήματος: $L=\max\{L(j): j=\{1,\dots,n\}\}$

Πολυπλοκότητα: Πλήθος Υποπροβλημάτων * Πλήθος επιλογών

$$\sum_{j=1}^{n} j-\cancel{1}+\cancel{1}=\sum_{j=1}^{n} j=\frac{n\cdot(n-1)}{2}=\Theta(n^2)$$

```
void maxsubseq (int A[], int parent[], int n) {
    for j=1 to n
        L[j] ← 0;
        parent[j] ← NULL;
    end-for                                  } Αρχικοποίηση

    for j=1 to n
        for i=1 to j-1
            if A[j] > A[i] and L[j] < L[i]
                L[j] ← L[i];
                parent[j] ← i;
            end-if
        end-for
        L[j]++;
    end_for                                  } Εύρεση της μακρύτερης αύξουσας υποακολουθίας
    length ← 0;
    last_node ← 0;
    for j=1 to n do
        if length < L[j]
            length ← L[j];
            last_node ← j;
        end_if
    end_for                                  } Βρίσκω το στοιχείο j με το μέγιστο L[j]
    while (last_node != NULL) do
        print (last_node);
        last_node ← parent[last_node];
    end-while                                } Τυπώνω την υποακολουθία
    print (length);
```

Θέμα Ιανουαρίου 2023

Αίτηση	f	j	b	g	a	h	i	d	c	e
j	1	2	3	4	5	6	7	8	9	10
Ημέρα Άφιξης (a_j)	2	2	3	4	5	6	7	7	8	11
Ημέρα Αναχώρησης (d_j)	4	6	6	6	7	12	9	14	11	12
~~$p(j)$~~	~~4~~	~~6~~	~~6~~	~~6~~	~~7~~	~~11~~	~~10~~	~~11~~	~~11~~	~~11~~
$L(j)$	2	4	3	2+2=4	2+2=4	6+4=10	2+4=6	7+4=11	3+4=7	1+7=8
$p(j)$	nil	nil	nil	f	f	g ή j	g ή a ή j	g ή a ή j	g ή a ή j	c

Γράψε τα 5 βήματα κ' Αλγόριθμο

1) Καθορισμός Υποπροβλημάτων: Ως υποπρόβλημα $L(i:j)$, όπου $j\in\{1,\dots,n\}$ και $i\in\{1,\dots,j-1\}$ ορίζω το πρόβλημα εύρεσης της μεγαλύτερης διάρκειας ενοικίασης του σπιτιού μεταξύ των αιτήσεων $i:j$

2) Καθορισμός Επιλογών: $L(j)=d(j)-a(j)+\max\{L(i): i\in\{1,\dots,j-1\}\}$

$\#$Επιλογών $= j-1-1+1 = j-1$

Πολυπλοκότητα: $\sum_{j=1}^{n} j-1 = \sum_{j=1}^{n} j - \sum_{j=1}^{n} 1 = \frac{n\cdot(n-1)}{2} - n = \frac{n^2}{2} - \frac{n}{2} - n \Rightarrow \Theta(n^2)$

3) Ορισμός Αναδρομικής Σχέσης: $L(j)=\begin{cases} d(j)-a(j), & j=1 \\ d(j)-a(j)+\max\{L(i): i\in\{1,\dots,j-1\}, d(i)\le a(j)\}, & n\ge j>1 \end{cases}$

4) Τοπολογική Ταξινόμηση:

```
for j=1 to n
    for i=1 to j-1
```

Όπου ο δείκτης j ακολουθεί τον αριθμό των αιτήσεων κατά αύξουσα σειρά

5) Επίλυση Αρχικού προβλήματος: Η εύρεση του μέγιστου $L(j)$, δηλαδή $\max\{L(j): j\in\{1,\dots,n\}\}$

Αλγόριθμος (Σπίτι)

Άσκηση)

Άσκηση 3 *Να περιγράψετε και να μελετήσετε αλγόριθμο ο οποίος δέχεται σαν είσοδο έναν πίνακα A με n διαφορετικά είδη (χαρτο)νομισμάτων σε κάποιο συναλλαγματικό μέσο (π.χ., ευρώ, δολάριο, κλπ) ταξινομημένα κατά αύξουσα σειρά και ένα χρηματικό ποσό X και υπολογίζει το X σαν άθροισμα με τον ελάχιστο αριθμό (χαρτο)νομισμάτων. Θεωρείστε ότι το μικρότερο (χαρτο)νόμισμα έχει αξία 1.*

$A[\,] = [1, 15, 25]$

1) Καθορισμός Υποπροβλημάτων: $p(x)$, όπου $x \in \{1, \dots, X\}$

$P(x)$ είναι το πρόβλημα εύρεσης του ελάχιστου πλήθους χαρτονομισμάτων για το ποσό $x \in \{1, \dots, X\}$

2) Καθορισμός Επιλογών: $i \in \{1, \dots, n\}$, $x - A[i] \geq 0$

$\#$Επιλογών $= n-1+1 = n$

3) Αναδρομική Σχέση:

$$p(x) = \begin{cases} 1 + \min\{p(x-A[i]) : x - A[i] \geq 0, i \in \{1, \dots, n\}\}, & X \geq x \geq 1 \\ 0, & x = 0 \end{cases}$$

4) Τοπολογική ταξινόμηση

for $x=1$ to X
  for $i=1$ to n

Ο δείκτης x ακολουθεί τις επιμέρους τιμές της αρχικής τιμής X, κατά αύξουσα σειρά

Πολυπλοκότητα: $\sum_{x=1}^{X} n = X \cdot n = \Theta(X \cdot n)$

5) Επίλυση Αρχικού Προβλήματος: $p(X)$