



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

2η Γραπτή Άσκηση

Αλγόριθμοι & Πολυπλοκότητα

Σπουδαστής:
Διονύσης ΖΗΝΔΡΟΣ (06601)
<dionyziz@gmail.com>

Διδάσκοντες:
Στάθης ΖΑΧΟΣ
Δημήτρης ΦΩΤΑΚΗΣ

31 Δεκεμβρίου 2011

Άσκηση 1

Άσκηση 2

Άσκηση 3

α)

Ένα παράδειγμα για το οποίο ο άπληστος αλγόριθμος δεν δίνει το βέλτιστο αποτέλεσμα είναι το εξής:

0	4999	0
4999	5000	4999
0	4999	0
0	0	0

Ενώ ο άπληστος αλγόριθμος θα επιλέξει να τοποθετήσει ένα βότσαλο στο κουτί με αξία 5000, η βέλτιστη λύση επιτυγχάνεται επιλέγοντας τα 4 κουτάκια που είναι γείτονές του και δίνουν συνολική αξία 19,996.

β)

Η λύση δυναμικού προγραμματισμού εντοπίζει τη βέλτιστη διάταξη σε γραμμικό χρόνο διασχίζοντας τη σκακιέρα από τα αριστερά προς τα δεξιά.

Σε κάθε βήμα, όταν επεξεργαζόμαστε την i -οστή στήλη, αποθηκεύουμε το βέλτιστο ποσό που μπορούμε να κερδίσουμε $J[i, \mu]$ από το μέρος της σκακιέρας που αποτελείται από τις στήλες 1 έως i για κάθε πιθανό συνδυασμό βοτσάλων μ που μπορούμε να τοποθετήσουμε στην i -οστή στήλη. Έχοντας αυτά τα δεδομένα, ο υπολογισμός της βέλτιστης λύσης για την $(i + 1)$ -οστή στήλη προκύπτει από την εξής αναδρομική σχέση:

$$\Omega = \{x \in 2^{\{1,2,3,4\}} : \forall \alpha, \beta \in x : \alpha - 1 \neq \beta\}$$

$$J[i, \mu] = \max\{J[i - 1, \mu'] + \sum_{q \in \mu} C[i, q] : \mu' \in \Omega \wedge \mu' \cap \mu = \emptyset\}$$

Algorithm 1 Άσκηση 3β

```

1: procedure CHESS( $C, n$ )
2:    $M \leftarrow \{x \in 2^{\{1,2,3,4\}} : \forall \alpha, \beta \in x : \alpha - 1 \neq \beta\}$ 
3:   for all  $\mu \in M$  do
4:      $J'[\mu] \leftarrow 0$ 
5:   for  $i \leftarrow 0$  to  $n$  do
6:     for  $\mu \in M$  do
7:        $J[\mu] \leftarrow 0$ 
8:       for  $\nu \in M$  do
9:         if  $\nu \cap \mu = \emptyset$  then
10:           $J[\mu] \leftarrow \max(J[\mu], J'[\nu] + \sum_{q \in \mu} C[i, q])$ 
11:        $J' \leftarrow J$ 
12:   return  $\max(\{J[\mu] : \mu \in M\})$ 

```

Άσκηση 4

Algorithm 2 Άσκηση 4

```

1: procedure LINESPLIT( $l, n, C$ )
2:    $J[0] \leftarrow 0$ 
3:   for  $i \leftarrow 1$  to  $n$  do
4:      $J[i] \leftarrow \infty$ 
5:      $cost \leftarrow C + 1$ 
6:     for  $j \leftarrow i$  downto 1 do
7:        $cost \leftarrow cost - (l[j] + 1)$ 
8:       if  $cost < 0$  then
9:         break
10:     $J[i] \leftarrow \min(J[i], J[j - 1] + cost^2)$ 
11:   return  $J[n]$ 

```

Άσκηση 5

Algorithm 3 Άσκηση 5

```

1: procedure SERVERS( $b, c, n$ )
2:    $\Omega \leftarrow \{i : b[i] > 0\}$ 
3:   if  $\Omega = \emptyset$  then
4:     return  $\emptyset$ 
5:    $K \leftarrow \max(\Omega)$ 
6:    $JS[K] \leftarrow c[K]$ 
7:    $J[K] \leftarrow c[K]$ 
8:    $W[K] \leftarrow K$ 
9:   for  $i \leftarrow K - 1$  downto 1 do
10:     $JS[i] \leftarrow J[i + 1] + C[i]$ 
11:     $J[i] \leftarrow \infty$ 
12:    for  $j \leftarrow i$  to  $W[i + 1]$  do
13:       $cost \leftarrow JS[j]$ 
14:      for  $l = i$  to  $j$  do
15:         $cost \leftarrow cost + (j - l)B[l]$ 
16:        if  $cost \geq J[i]$  then
17:          break
18:        if  $cost < J[i]$  then
19:           $J[i] \leftarrow cost$ 
20:           $W[i] \leftarrow j$ 
21:    $S \leftarrow \{S_K\}$ 
22:    $i \leftarrow W[1] + 1$ 
23:   while  $i \leq K$  do
24:      $S \leftarrow S \cup \{S_{i-1}\}$ 
25:      $i \leftarrow W[i] + 1$ 
26:   return  $S$ 

```

Άσκηση 6

α)

β)

γ)