

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή ΗΜ&ΜΥ
Αλγόριθμοι και Πολυπλοκότητα
7^ο εξάμηνο, Ροή Λ
Ακαδημαϊκή περίοδος: 2010-2011



2^η Σειρά Γραπτών Ασκήσεων

Γερακάρης Βασίλης
<vgerak@gmail.com>
Α.Μ.: 03108092

2 Ιανουαρίου 2012

1 Επιτροπή Αντιπροσώπων (KT 4.15)

Ταξινομούμε τον πίνακα σε αύξουσα σειρά με βάση τα f_i . Βρίσκουμε το διάστημα (έστω m) με το μέγιστο f_m που η αρχή του s_m βρίσκεται πριν το τέλος του 1ου f_1 και το επιλέγουμε ως αντιπρόσωπο. Θα αποδείξουμε ότι η επιλογή που κάνουμε (σε κάθε βήμα) είναι και η βέλτιστη δυνατή.

Έστω k , ένας διαφορετικός αντιπρόσωπος, που είναι η βέλτιστη επιλογή. Δεν είναι δυνατόν να έχει δείκτη $k > m$, αφού έτσι δε θα επικάλυπτε το 1ο διάστημα (εξ'ορισμού του m). Αφού λοιπόν $k \leq m$ θα ισχύει και $f_k \leq f_m$. Άρα ο k επικαλύπτει **το πολύ** όσα διαστήματα επικαλύπτει η επιλογή μας, επομένως η επιμέρους λύση μας είναι η βέλτιστη. Επαγωγικά προκύπτει η ορθότητα του αλγορίθμου μας.

Algorithm 1 Άσκηση 1

```
1: Sort A on ascending order using  $f_i$  as key
2: procedure RepresentativesSelect( $A, N$ )
3:   if Empty( $A$ ) then
4:     return  $RepresentativesList$ 
5:   else
6:      $i = 1$ 
7:     while  $i < N$  and  $s_i < f_1$  do
8:        $i \leftarrow i + 1$ 
9:      $i \leftarrow i - 1$ 
10:     $RepresentativesList.append(i)$ 
11:    if  $f_i > s_{i+1}$  then
12:       $A.remove(i + 1)$ 
13:       $N \leftarrow N - 1$ 
14:    for  $j \leftarrow i$  to 1 step -1 do
15:       $A.remove(j)$ 
16:       $N \leftarrow N - 1$ 
17:    return RepresentativesSelect ( $A, N$ )
```

2 Βιαστικός Μοτοσυκλετιστής

Ταξινομούμε τα διαστήματα σε αύξουσα σειρά με βάση τα όρια ταχυτήτων v_i και δημιουργούμε τον πίνακα χρόνων t , όπου $t_i = l_i/v_i$. Έστω T ο χρόνος που θα ξεπεράσουμε το όριο ταχύτητας και u τα χιλιόμετρα κατά τα οποία θα το ξεπεράσουμε. Εάν $T \leq t_1$, τότε κάνουμε T λεπτά με $v_1 + u$ km/h, και τα υπόλοιπα με το όριο ταχύτητας, αλλιώς κάνουμε t_1 λεπτά υπερβαίνοντας το όριο και επαναλαμβάνουμε τη διαδικασία για $T - t_1$ λεπτά για το επόμενο διάστημα.

Θα αποδείξουμε ότι η επιλογή που κάνουμε σε κάθε βήμα, είναι η βέλτιστη δυνατή. Ο συνολικός χρόνος που χρειάζεται για να φτάσει στον προορισμό του είναι:

$$T_{total} = \sum_{i=1}^n t_i = \sum_{i=1}^n \frac{l_i}{u_i}.$$

Η μέγιστη (ποσοστιαία) ελάττωση των χρονικών διαστημάτων προκύπτει αν προσθέσουμε την ταχύτητα u στο μικρότερο παρονομαστή v_i . Καθώς λοιπόν όλες οι τιμές είναι δεδομένες και σταθερές (T, v_i, l_i) , οποιοδήποτε άλλο διάστημα αν επιλέγαμε, θα έδινε ποσοστιαίο κέρδος χρόνου **το πολύ** όσο αυτό που υπολογίσαμε, επομένως η επιμέρους λύση μας είναι η βέλτιστη.

Επαγωγικά προκύπτει η ορθότητα του αλγορίθμου μας.

Algorithm 2 Άσκηση 2

```
1: Sort A on ascending order using  $v_i$  as key
2: for  $i \leftarrow 1$  to  $N$  do
3:    $t_i \leftarrow l_i/v_i$ 
4: procedure TimeSelect( $A, N$ )
5:    $i \leftarrow 1$ 
6:   while  $T \geq t_i$  do
7:      $Selection.append(i, t_i)$ 
8:      $T \leftarrow T - t_i$ 
9:      $i \leftarrow i + 1$ 
10:   $Selection.append(i, T)$ 
11:  return  $Selection$ 
```

3 Βότσαλα στη Σκακιέρα (DVP 6.5)

Test3

4 Χωρισμός Κειμένου σε Γραμμές

Test4

5 Αντίγραφα Αρχείου (KT 6.12)

Test5

6 Bonus: Έλεγχος Ταξινόμησης

Bonus!