Αλγόριθμοι και Πολυπλοκότητα
2η Σειρά Γραπτών Ασκήσεων

Γιαννέλος Γιάννης
ΑΜ:03108088

2 Ιανουαρίου 2012

Άσκηση 2: Βιαστικός μοτοσυκλετιστής

Παραβίαση ορίου ταχύτητας κατα $v_i + V$

Στην περίπτωση που ο μοτοσυκλετιστής αποφασίζει να παραβιάσει το εκάστοτε όριο ταχύτητας σε κάθε τμήμα του δρόμου που συνδέει την πόλη i με την πόλη $i+1$ κατά σταθερά ταχύτητα V, ένας greedy αλγόριθμος θα ήταν ο εξής:

- Έστω πίνακας $X[n]$ οπου $X[i] = l_1$ για $i = 1...n$ (πίνακας των αποστάσεων μεταξύ των πόλεων).
- Έστω πίνακας $V_{max}[n][2]$ όπου κάθε στοιχείο του $V_{max}[i][1]$ για $i = 1...n$ είναι η επιτρεπόμενη ταχύτητα για κάθε i αν προσθέσουμε σε αυτή το V. Δηλαδή $V_{max}[i][1] = v_1 + V$. Στην δεύτερη στήλη αποθηκεύουμε το i.
- Ταξινομούμε τον πίνακα V_{max} σε φθίνουσα σειρά σύμφωνα με την 1η στήλη.
- Η βέλτιστη κατανομή του χρόνου που ο μοτοσυκλετιστής παραβιάζει το όριο ταχύτητας θα προκύψει αν εξαντλήσουμε χρονικά τα τμήματα με την μεγαλύτερη επιτρεπτή ταχύτητα.

Ακολουθεί ψευδοκώδικας για την επίλυση του προβλήματος:

Algorithm 1 Βιαστικός Μοτοσυκλετιστής

```
Initialize t_optimal[n] ← 0          ▷ Αρχικοποίηση πίνακα t_optimal μεγέθους n
i ← 1
for i = 1 → n do
    if T ≥ V[i]/X[i] then
        t_optimal[i] ← V[i]/X[i]
        i ← i + 1
        T ← T − t_optimal[i]
    else
        break
    end if
end for
```

Άσκηση 4: Χωρισμός κειμένου σε γραμμές

Όπως ορίζεται και στην εκφώνηση, το κείμενο που θέλουμε να χωρίσουμε αποτελείται απο n λέξεις μήκους $l_1, l_2, ..., l_n$ χαρακτήρες η κάθε μια. Οι γραμμές θα πρέπει να έχουν μέγεθος το πολύ C χαρακτήρες και κάθε γραμμή στοιχίζεται στα

αριστερά. Ακόμα κάθε λέξη χωρίζεται απο την επόμενη με τον κενό χαρακτήρα χωρίς να επιτρέπεται να χωριστεί σε 2 γραμμές και δεν επιτρέπεται η αναδιάταξη τους. Άρα συνολικά, αν στη γραμμή k εμφανίζονται οι λέξεις $i, ..., j$ τότε η γραμμή έχει αριθμό κενών χαρακτήρων στα δεξιά που δίνεται απο τον τύπο $s_k = C+1-\sum_{p=1}^{j}(l_p+1)$. Το ζητούμενο της άσκησης είναι να βρούμε αλγόριθμο που να χωρίζει κατάλληλα το κείμενο έτσι ώστε να ελαχιστοποιείται το άθροισμα των τετραγώνων του πλήθους των κενών χαρακτήρων που έχουν οι γραμμές στα δεξιά, δηλαδή το $\sum_{k=1}^{m} s_k^2$. Για την επίλυση θα εργαστούμε με την μέθοδο σχεδίασης bottom-up χρησιμοποιώντας δυναμικό προγραμματισμό:

- Η κύρια ιδέα στηρίζεται στο ότι αν μπορούμε να υπολογίσουμε το ελάχιστο $\sum_{k=1}^{\lambda} s_k^2$ για τις πρώτες λ σειρές, τότε αν προσθέσουμε τις λέξεις για την $\lambda+1$ σειρά, δεν αλλάζει κάτι στην στοίχιση των προηγούμενων σειρών παρα μόνο της τελευταίας.

- Υπολογίζουμε τον πίνακα S όπου για κάθε i, j το στοιχείο $S[i,j]$ δείχνει το $s_k = C+1-\sum_{p=1}^{j}(l_p+1)$.

- Με την παραπάνω ιδέα είναι εφικτό να φτιάξουμε μια αναδρομική σχέση για το "κόστος" της διάταξης των πρώτων i λέξεων του κειμένου . Η σχέση αυτή θα είναι:

$$C[i] = \begin{cases} 0, & i = 0 \\ \min\limits_{1 \le k \le i} C[k-1] + S[k,i], & i > 0 \end{cases}$$

Η πολυπλοκότητα του αλγορίθμου για τον υπολογισμό του πίνακα S είναι $O(n \cdot C)$ όπου n το πλήθος των λέξεων και C το μέγιστο πλήθος χαρακτήρων ανά γραμμή. Το ίδιο ισχύει και για τον υπολογισμό του $C[n]$.