



16^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 2004 ΘΕΜΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Πρόβλημα Αποθήκευση Προγραμμάτων

Περιγραφή Προβλήματος

Μόλις προσληφθήκατε σε μια εταιρεία ανάπτυξης λογισμικού και ο προϊστάμενος σας ανέθεσε το ακόλουθο πρόβλημα. Να αποθηκευθούν n προγράμματα σε k σκληρούς δίσκους ενός υπολογιστή. Δίνεται ότι η απαιτούμενη μνήμη για αποθήκευση του i προγράμματος είναι l_i , $i = 1, 2, \dots, n$, η δε χωρητικότητα του κάθε δίσκου σε μνήμη είναι L . Κάθε πρόγραμμα πρέπει να αποθηκευθεί ολόκληρο σε ένα δίσκο και απαιτεί μνήμη μικρότερη ή ίση της χωρητικότητας του κάθε δίσκου. Επίσης, η συνολική μνήμη των n προγραμμάτων υπερβαίνει την συνολική χωρητικότητα των k δίσκων.

1) Ζητούμενο (k δίσκοι) [20 μονάδες]

Να γραφεί πρόγραμμα, το οποίο να δέχεται σαν δεδομένα τα n , L , k και l_i , $i = 1, 2, \dots, n$ και να υπολογίζει και τυπώνει το μέγιστο πλήθος από τα n προγράμματα, τα οποία μπορούν να αποθηκευθούν στους k δίσκους καθώς και τις αντίστοιχες ποσότητες μνήμης τους. Ο χρόνος εκτέλεσης του προγράμματός σας να είναι ο ελάχιστος δυνατός διότι θα αποτελέσει βασικό κριτήριο στη βαθμολόγηση.

Είσοδος Δεδομένων

Η είσοδος του προγράμματός σας δίνεται από ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. INPUT1.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει τρεις ακέραιους αριθμούς n , L και k , όπου n το πλήθος των προγραμμάτων, L η χωρητικότητα σε μνήμη ενός δίσκου και k το πλήθος των δίσκων.
- Η κάθε μία από τις επόμενες n γραμμές περιέχει ένα ακέραιο αριθμό, ο οποίος αντιστοιχεί στην χωρητικότητα μνήμης ενός προγράμματος. Η πρώτη γραμμή περιέχει την χωρητικότητα μνήμης του πρώτου προγράμματος (l_1), η δεύτερη γραμμή την χωρητικότητα μνήμης του δεύτερου προγράμματος (l_2) κ.ο.κ.

Έξοδος Αποτελεσμάτων

Η έξοδος του προγράμματός σας θα πρέπει να δίνεται σε ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. OUTPUT1.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει $k+1$ ακέραιους αριθμούς, ο πρώτος είναι το μέγιστο πλήθος από τα n προγράμματα, τα οποία μπορούν να αποθηκευθούν στους k δίσκους, ο δεύτερος είναι το πλήθος των προγραμμάτων που αποθηκεύθηκαν στον πρώτο δίσκο, ο τρίτος είναι το πλήθος των προγραμμάτων που αποθηκεύθηκαν στο δεύτερο δίσκο κ.ο.κ.



- Η κάθε μία από τις επόμενες γραμμές περιέχει τις χωρητικότητες μνήμης των προγραμμάτων που αποθηκεύθηκαν σε κάθε δίσκο. Πρώτα γράφονται οι χωρητικότητες μνήμης των προγραμμάτων που αποθηκεύθηκαν στον πρώτο δίσκο και ακολουθούν αυτές που αποθηκεύθηκαν στο δεύτερο δίσκο κ.ο.κ.

Παράδειγμα Εισόδου

8 15 3
11
7
8
3
5
12
4
5

Παράδειγμα Εξόδου

6 1 2 3
11
5 7
4 3 5

Παρατήρηση

- 1) Όπως μπορείτε εύκολα να διαπιστώσετε η ανωτέρω λύση δεν είναι μοναδική και δεν είναι βέλτιστη.
- 2) Μην προσπαθήσετε να βρείτε τη βέλτιστη λύση, αλλά μια όσο το δυνατόν καλή λύση πλησίον της βέλτιστης.

Αυτοαξιολόγηση

Για να διαπιστώσετε την ορθότητα και αποτελεσματικότητα του προγράμματός σας μπορείτε να χρησιμοποιήσετε σαν αρχεία εισόδου τα: inpar1_1.txt, inpar1_2.txt και inpar1_3.txt.

Οι αντίστοιχες λύσεις βρίσκονται στα αρχεία: outpar1_1.txt, outpar1_2.txt και outpar1_3.txt.

2) Ζητούμενο (ελάχιστοι δίσκοι) [30 μονάδες]

Αν η χωρητικότητα μνήμης κάθε προγράμματος είναι μεγαλύτερη από $L/3$, όπου L η χωρητικότητα μνήμης κάθε δίσκου, τότε να γραφεί ένα πρόγραμμα, το οποίο να τοποθετεί και τα n προγράμματα στο ελάχιστο πλήθος δίσκων.

Είσοδος Δεδομένων

Η είσοδος του προγράμματός σας δίνεται από ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. INPUT2.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει δύο ακέραιους αριθμούς n και L , όπου n το πλήθος των προγραμμάτων και L η χωρητικότητα σε μνήμη κάθε δίσκου.
- Η κάθε μία από τις επόμενες n γραμμές περιέχει ένα ακέραιο αριθμό, ο οποίος αντιστοιχεί στην χωρητικότητα μνήμης ενός προγράμματος. Η πρώτη γραμμή περιέχει την χωρητικότητα μνήμης του πρώτου προγράμματος (l_1), η δεύτερη γραμμή την χωρητικότητα μνήμης του δεύτερου προγράμματος (l_2) κ.ο.κ.



Έξοδος Αποτελεσμάτων

Η έξοδος του προγράμματός σας δίνεται σε ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. OUTPUT2.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει ένα ακέραιο αριθμό που είναι το ελάχιστο πλήθος δίσκων.
- Η κάθε μία από τις επόμενες γραμμές περιέχει τις χωρητικότητες μνήμης των προγραμμάτων που αποθηκεύθηκαν σε κάθε δίσκο. Πρώτα γράφονται οι χωρητικότητες μνήμης των προγραμμάτων που αποθηκεύθηκαν στον πρώτο δίσκο και ακολουθούν αυτές που αποθηκεύθηκαν στο δεύτερο δίσκο κ.ο.κ.

Παράδειγμα Εισόδου

6 7
3
5
3
4
3
3

Παράδειγμα Εξόδου

4
4 3
5
3 3
3

Αυτοαξιολόγηση

Για να διαπιστώσετε την ορθότητα και αποτελεσματικότητα του προγράμματός σας μπορείτε να χρησιμοποιήσετε σαν αρχεία εισόδου τα: inpar2_1.txt, inpar2_2.txt και inpar2_3.txt.

Οι αντίστοιχες λύσεις βρίσκονται στα αρχεία: outpar2_1.txt, outpar2_2.txt και outpar2_3.txt.

ΠΡΟΣΟΧΗ

1. Το όνομα του εκτελέσιμου προγράμματος που θα δημιουργήσετε πρέπει να είναι **apoth1.exe** και **apoth2.exe** για κάθε ζητούμενο, αντίστοιχα.
2. Γράφετε όσο το δυνατόν πιο ευανάγνωστα τον κώδικα του προγράμματός σας και τοποθετείτε σχόλια.

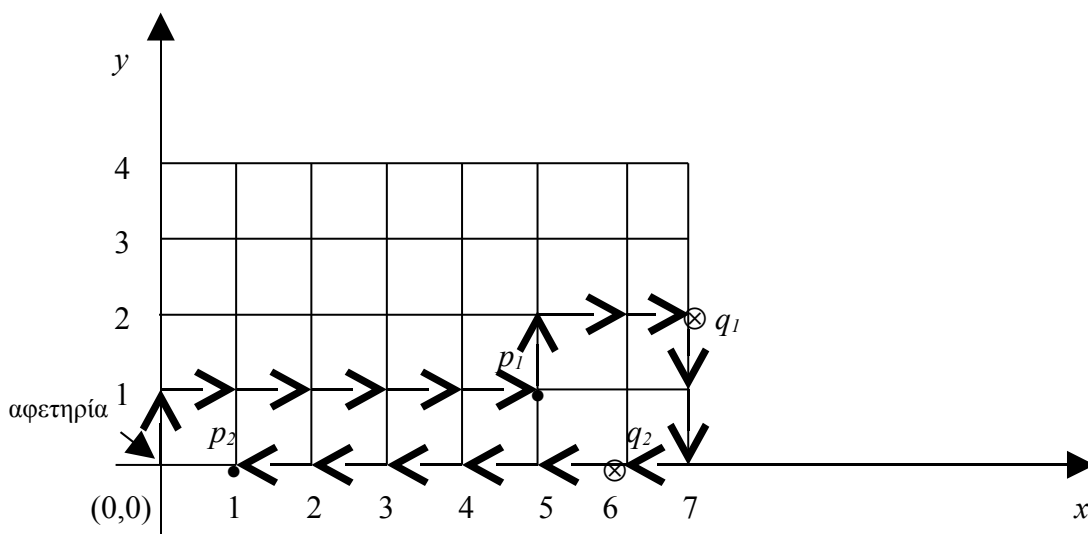
Πρόβλημα Ο Αρχαιολογικός χώρος

Περιγραφή Προβλήματος

Το σχέδιο ενός αρχαιολογικού χώρου είναι ένα πλέγμα (βλ. Σχήμα) όπου οι γραμμές παριστάνουν διαδρόμους. Τα αρχαιολογικά ευρήματα βρίσκονται στα σημεία τομής των διαδρόμων. Υπάρχουν δύο ακολουθίες ευρημάτων (με ακέραιες συντεταγμένες) η $P = \{p_1, p_2, \dots, p_n\}$, και η $Q = \{q_1, q_2, \dots, q_m\}$, όπου p_i, q_i συμβολίζουν το i -ιοστό εύρημα της P και Q ακολουθίας,



αντίστοιχα. Ένας ξεναγός, μαζί με την ομάδα (γκρούπ) του επιθυμεί να επισκεφθεί διαμέσου των διαδρόμων όλα τα ευρήματα των δύο ακολουθιών ξεκινώντας από το σημείο $(0,0)$. Για ιστορικούς λόγους τα ευρήματα της κάθε ακολουθίας πρέπει να επισκεφθούν με την σειρά που εμφανίζονται στην ακολουθία. Πιο συγκεκριμένα, ο ξεναγός επιθυμεί να επισκεφθεί διαμέσου των διαδρόμων το εύρημα p_{i-1} πριν από το p_i , για $i = 2, 3, \dots, n$ και το εύρημα q_{j-1} πριν από το q_j για $j = 2, 3, \dots, m$. Επειδή, η κίνηση του ξεναγού με την ομάδα του γίνεται μόνο διαμέσου των διαδρόμων (οριζόντια / κατακόρυφα) στον αρχαιολογικό χώρο, η απόσταση μεταξύ δύο αρχαιολογικών ευρημάτων με συντεταγμένες (x_1, y_1) και (x_2, y_2) είναι $|x_1 - x_2| + |y_1 - y_2|$.



Σχήμα: Με $\bullet$ παριστάνονται τα ευρήματα της ακολουθίας P και με $\otimes$ τα ευρήματα της Q . Η διαδρομή με τα βέλη είναι η μικρότερη δυνατή απόσταση.

3) Ζητούμενο [50 μονάδες]

Να γραφεί ένα πρόγραμμα, το οποίο να δέχεται ως δεδομένα τις συντεταγμένες των αρχαιολογικών ευρημάτων δύο ακολουθιών και να υπολογίζει και τυπώνει τη μικρότερη δυνατή απόσταση που θα διανύσει ένας ξεναγός με την ομάδα του για να επισκεφθεί όλα τα αρχαιολογικά ευρήματα με τον τρόπο που περιγράφηκε ανωτέρω. Ο χρόνος εκτέλεσης του προγράμματός σας να είναι ο ελάχιστος δυνατός διότι θα αποτελέσει βασικό κριτήριο στη βαθμολόγηση.

Είσοδος Δεδομένων

Η είσοδος του προγράμματός σας δίνεται από ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. INPUT3.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει δύο ακέραιους αριθμούς n και m , όπου n ($1 \leq n \leq 2047$) είναι το πλήθος των ευρημάτων της ακολουθίας P και m ($1 \leq m \leq 2047$) είναι το πλήθος των ευρημάτων της ακολουθίας Q .



- Η κάθε μία από τις επόμενες n γραμμές περιέχει δύο ακέραιους αριθμούς στο διάστημα $[-2047, 2047]$, που είναι η x και y συντεταγμένη ενός ευρήματος της P ακολουθίας και χωρίζονται με ένα τουλάχιστον κενό.
- Η κάθε μια από τις επόμενες m γραμμές περιέχουν δύο ακέραιους αριθμούς στο διάστημα $[-2047, 2047]$, που είναι η x και y συντεταγμένη ενός ευρήματος της Q ακολουθίας και χωρίζονται με ένα τουλάχιστον κενό.

Έξοδος Αποτελεσμάτων

Η έξοδος του προγράμματός δίνεται σε ένα αρχείο καθοριζόμενο από τον χρήστη (π.χ. OUTPUT3.TXT) και αποτελείται από τις εξής γραμμές:

- Η πρώτη γραμμή περιέχει ένα ακέραιο αριθμό, ο οποίος είναι η ελάχιστη απόσταση που θα διανύσει ο ξεναγός με την ομάδα του.
- Η κάθε μία από τις επόμενες $n+m$ γραμμές περιέχει δύο ακέραιους αριθμούς που είναι οι συντεταγμένες των ευρημάτων με τη σειρά που τα επισκέφθηκε ο ξεναγός.

Παράδειγμα Εισόδου

2 2
5 1
1 0
7 2
6 0

Παράδειγμα Εξόδου

17
5 1
7 2
6 0
1 0

Παρατήρηση

- 1) Ο υπολογισμός της ελάχιστης απόστασης, ξεκινώντας από το $(0,0)$ (βλ. βέλη στο Σχήμα) γίνεται ως εξής $6+3+3+5 = 17$. Εδώ υποθέτουμε ότι η απόσταση μεταξύ δύο γειτονικών σημείων του πλέγματος είναι 1.
- 2) Μία άλλη πιθανή διαδρομή είναι η $(5,1), (1,0), (7,2), (6,0)$ με συνολική απόσταση $6+5+8+3 = 22$, η οποία όμως δεν είναι η ελάχιστη δυνατή.

Αυτοαξιολόγηση

Για να διαπιστώσετε την ορθότητα και αποτελεσματικότητα του προγράμματός σας μπορείτε να χρησιμοποιήσετε σαν αρχεία εισόδου τα: `inpar3_1.txt` και `inpar3_2.txt`.

Οι αντίστοιχες λύσεις βρίσκονται στα αρχεία: `outpar3_1.txt` και `outpar3_2.txt`.

ΠΡΟΣΟΧΗ:

1. Το όνομα του εκτελέσιμου προγράμματος που θα δημιουργήσετε πρέπει να είναι **arxaios.exe**.
2. Γράφετε όσο το δυνατόν πιο ευανάγνωστα τον κώδικα του προγράμματός σας και τοποθετείτε σχόλια.

Καλή Επιτυχία