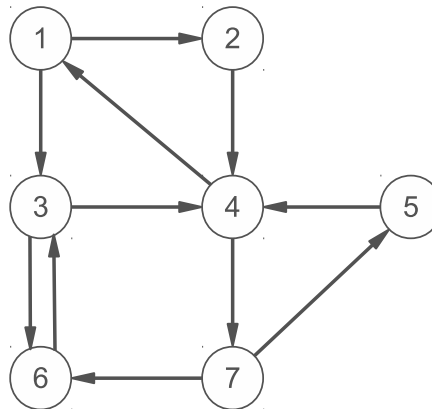


Αλγόριθμοι και Πολυπλοκότητα
Προγραμματιστική Άσκηση
Παράδοση: Παρασκευή 5 Ιουνίου 2020

Πρόβλημα 1.

Υλοποιήστε τον γνωστό αλγόριθμο από την ύλη του μαθήματος που ελέγχει εάν ένα κατευθυνόμενο γράφημα G είναι ισχυρά συνδεδεμένο ή όχι.

Το γράφημα εισόδου πρέπει να διαβάζεται από ένα αρχείο με όνομα `graph.in` που έχει την ακόλουθη μορφή. Στην πρώτη γραμμή του αρχείου δίνεται το πλήθος των κόμβων n . Θεωρήστε ότι οι κόμβοι είναι αριθμημένοι από 1 έως n . Στη συνέχεια ακολουθούν n γραμμές. Στην i -οστή γραμμή περιέχονται στη σειρά οι γειτονικοί κόμβοι του κόμβου i . Συγκεκριμένα ο πρώτος ακέραιος d αντιστοιχεί στο βαθμό του κόμβου i και οι επόμενοι d ακέραιοι στους γειτονικούς του κόμβους.



Για παράδειγμα, ένα αρχείο εισόδου που αντιστοιχεί στο γράφημα του σχήματος είναι το εξής:

```
7
2 2 3
1 4
2 4 6
2 1 7
1 4
1 3
2 5 6
```

Το πρόγραμμα πρέπει να διαβάζει και να αποθηκεύει το G στη μνήμη σε χώρο $O(m + n)$ όπου m το πλήθος των ακμών και n το πλήθος των κόμβων, να εκτελεί τον αλγόριθμο, να εκτυπώνει στην οθόνη εάν το G είναι ισχυρά συνδεδεμένο ή όχι και κατόπιν να τερματίζει. Ο συνολικός χρόνος εκτέλεσης θα πρέπει να είναι $O(m + n)$.

Δίνουμε μία ενδεικτική έξοδο την οποία το πρόγραμμα θα μπορούσε να παράγει με είσοδο το παραπάνω γράφημα:

Nai, to grafhma einai isxyra syndedemenο.

Πρόβλημα 2.

Να υλοποιήσετε *άπλοστο* αλγόριθμο για το πρόβλημα του 0-1 σακιδίου. Το πρόγραμμα σας θα πρέπει να διαβάζει από την τυπική είσοδο σε δύο διαφορετικές γραμμές το βάρος του σακιδίου W και το πλήθος των αντικειμένων n . Στη συνέχεια θα πρέπει να διαβάζει n γραμμές όπου στην i -στη γραμμή δίνεται το βάρος και η αξία του i -στου αντικειμένου, χωρισμένα με ένα κενό. Αφού διαβάσει την είσοδο θα εκτελεί τον αλγόριθμο και θα τυπώνει τη λύση που υπολογίζει δηλαδή την αξία του βέλτιστου σακιδίου και το σύνολο των αντικειμένων που αυτό περιέχει. Για παράδειγμα, ένα αρχείο εισόδου θα μπορούσε να είναι το εξής:

```
60
5
5 30
10 20
20 100
30 90
40 160
```

Για την παραπάνω είσοδο μία ενδεικτική έξοδο την οποία το πρόγραμμα θα μπορούσε να παράγει είναι η ακόλουθη:

```
Axia beltistou sakidiou: 260
Antikeimena: 3, 5
```

Πρόβλημα 3.

Να υλοποιήσετε τον αλγόριθμο *δυναμικού προγραμματισμού* χρόνου $O(n^3)$ ο οποίος υπολογίζει το ελάχιστο κόστος πολλαπλασιασμού μίας αλυσίδας πινάκων.

Το πρόγραμμα θα πρέπει να δέχεται ένα όρισμα που αντιστοιχεί στο όνομα του αρχείου εισόδου. Το αρχείο εισόδου περιέχει τις διαστάσεις των πινάκων της αλυσίδας πινάκων με την ακόλουθη μορφή. Στην πρώτη γραμμή του αρχείου δίνεται το πλήθος των πινάκων n . Στην συνέχεια ακολουθούν $n + 1$ γραμμές. Στην j -στη γραμμή δίνεται το πλήθος των γραμμών του j -στου πίνακα, για $1 \leq j \leq n$. Στην τελευταία γραμμή δίνεται το πλήθος των στηλών του n -στου πίνακα. Οι $n + 1$ παραπάνω γραμμές περιέχουν τις τιμές p_i , για $0 \leq i \leq n$.

Αφού το πρόγραμμα διαβάσει το αρχείο εισόδου εκτελεί τον αλγόριθμο δυναμικού προγραμματισμού και τυπώνει στην οθόνη το ελάχιστο κόστος πολλαπλασιασμού. Εφόσον $n \leq 15$, το πρόγραμμα θα πρέπει να τυπώνει επιπλέον και τον πίνακα M με το ελάχιστο κόστος για κάθε υποαλυσίδα πινάκων. Εφόσον $n \leq 100$, θα πρέπει να τυπώνει στην οθόνη σε μία γραμμή και μία βέλτιστη σειρά υπολογισμού του γινομένου των πινάκων. Πχ. αν για μία είσοδο πέντε πινάκων μία βέλτιστη σειρά υπολογισμού του γινομένου είναι $(A_1 A_2)((A_3 A_4) A_5)$ θα τυπώνει $(1\ 2)\ ((3\ 4)\ 5)$.