

ΕΡΩΤΗΜΑ 11

(+ερώτημα 12)

Στην άσκηση 11 υλοποιούνται κάποιες από τις βασικές συναρτήσεις των γράφων (μη-κατευθυνόμενων) με βάρη.

ask11.c

Είναι το αρχείο που περιλαμβάνει μόνο τη main() συνάρτηση. Ομοίως με την ΑΣΚΗΣΗ 9 & 10 και όπως εξηγώ στο αντίστοιχο pdf αρχείο χρησιμοποιώ μια μεταβλητή τύπου struct graph που την ονομάζω “graph” και έναν παράλληλο πίνακα ver[MAXVERTEX]. Επίσης χρησιμοποιώ έναν πίνακα edge[] μέσα στον οποίο αποθηκεύονται οι ακμές με τη σειρά που τις δίνει ο χρήστης και έναν πίνακα wght[] (wght ~ weight) παράλληλο του edge[], στον οποίο αποθηκεύονται τα βάρη των ακμών.

ask11mods.c

Είναι το αρχείο που περιλαμβάνει όλες τις απαιτούμενες συναρτήσεις για να εκτελεστούν οι ζητούμενες λειτουργίες.

Initialize / InsertEdge / ShowGraph ~ GRAPHinit / GRAPHedge / showGRAPH :

- Οι παραπάνω συναρτήσεις εκτελούν ακριβώς τις ίδιες λειτουργίες με αυτές της ΑΣΚΗΣΗΣ 10 με τη διαφορά ότι στον ορισμό του struct edge έχει προστεθεί και το (int) weight που είναι το βάρος της ακμής x-y.

MinimumSpanningTree (MinimumSpanningTree) :

- Είναι μια void συνάρτηση που εμφανίζει το ελάχιστο δέντρο επικάλυψης του γράφου που δέχεται ως όρισμα. Η συνάρτηση υλοποιεί τον αλγόριθμο Prim-Jarnik και γίνεται με adjacency list searching (χωρίς χρήση heap).

***.h αρχεία**

Είναι τα header files που χρησιμοποιούνται στα *.c αρχεία.

- ask11.h: περιέχει τις δηλώσεις των συναρτήσεων
- GraphTypes.h: περιέχει ορισμούς κάποιων struct & κάποια typedef

Το makefile μου έχει τις εξής λειτουργίες:

- make
 - gcc -c ask11.c
 - gcc -c ask11mods.c
 - gcc -o ask11 ask11.o ask11mods.o
- make clean
 - rm *.o ask11

Για την εκτέλεση του προγράμματος αρκεί να γράψετε:

- make
- ./ask11 < file.txt

(Όπου file.txt είναι τα .txt αρχεία που θέλετε να τεστάρετε)

Υ.Γ.: Μέσα σε κάθε φάκελο υπάρχουν και μερικά .txt αρχεία που χρησιμοποιήθηκαν για τον έλεγχο των προγραμμάτων μου.

ΑΣΚΗΣΗ 12

Ο αλγόριθμος Prim-Jarnik για την εύρεση του Minimum Spanning Tree, από τη στιγμή που υλοποιείται μόνο με adjacency list searching, και δε χρησιμοποιεί ούτε κάποιο heap ούτε επεξεργάζεται τα edges, η πολυπλοκότητά του είναι $O(n^2)$, όπου: “n” είναι το πλήθος των κορυφών του γράφου.

Υ.Γ.: Έγινε ερώτηση στο Piazza για το αν μπορώ να υλοποιήσω τον αλγόριθμο και χωρίς heap και μου είπαν ότι δεν υπάρχει πρόβλημα. (Οπότε ουσιαστικά δεν ακολούθησα πιστά τον αλγόριθμο των διαφανειών).