



Εργαστηριακές Σημειώσεις - Εβδομάδα 12

Σε αυτό το εργαστήριο θα μελετηθούν:

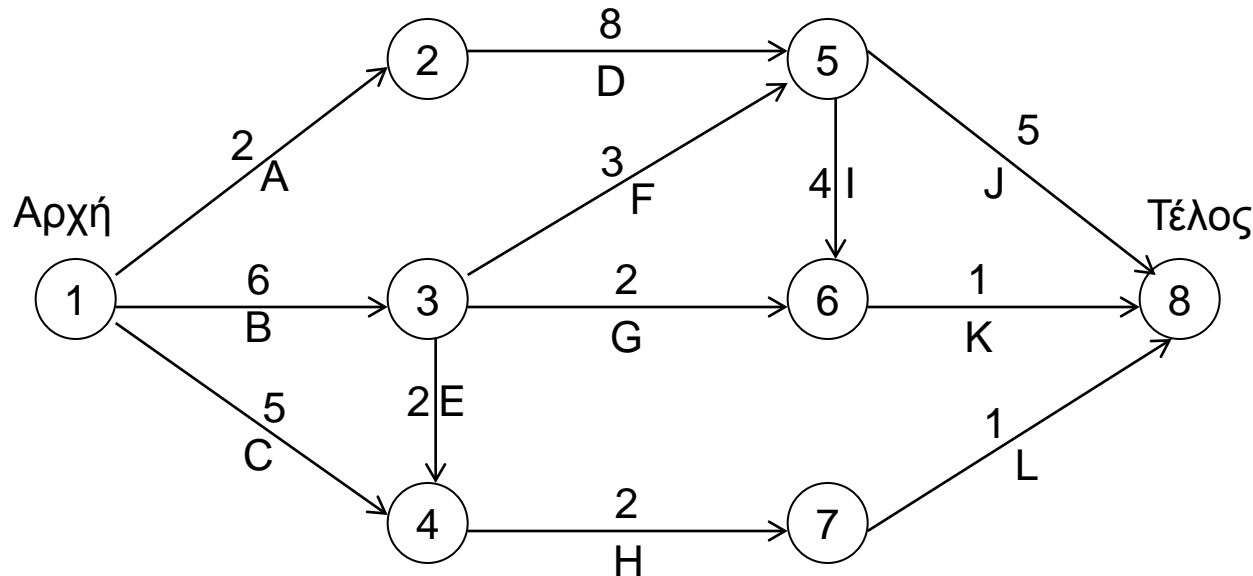
- Η μέθοδος κρίσιμης διαδρομής (*critical path method (CPM)*)
 - Χρησιμοποίηση της *CPM* για την ανάλυση κρίσιμων πτυχών ενός πολύπλοκου έργου
- Υλοποίηση *Topological Sort* και *Critical Path Method (CPM)*

Critical Path Method (CPM)



- Η μέθοδος κρίσιμης διαδρομής χρησιμοποιείται για τη διαχείριση πολύπλοκων έργων
 - Ο διαχειριστής του έργου δημιουργεί ένα δίκτυο από όλες τις φάσεις του έργου και στη συνέχεια αξιολογεί το δίκτυο για να καθορίσει τα κρίσιμα σημεία του έργου
- Η διαδικασία CPM μπορεί να μας βοηθήσει να καθορίσουμε:
 - Το χρόνο που θα χρειαστεί για την ολοκλήρωση του έργου
 - Ποιες δραστηριότητες είναι «κρίσιμες»
 - Τυχών καθυστέρηση κρίσιμης διεργασίας συνεπάγεται καθυστέρηση του όλου έργου
- Σε ένα δίκτυο CPM
 - Κάθε κόμβος (vertex) αντιστοιχεί σε ένα γεγονός (milestone), όπως η αρχή ή το τέλος μιας δραστηριότητας
 - Μια ακμή η οποία συνδέει δύο κόμβους αναπαριστά τη χρονική διάρκεια της δραστηριότητας

CMP για το χτίσιμο ενός σπιτιού



- Θα μπορούσαμε να εξάγουμε το πιο σύντομο μονοπάτι από την αρχή ως το τέλος για το πιο πάνω έργο, αλλά αυτό δεν θα είχε σημαντική αξία.
- Από την άλλη αν καθορίσουμε το μέγιστο μονοπάτι (το μονοπάτι με το μεγαλύτερο άθροισμα βαρών) θα είχαμε γνώση για το ποιες δραστηριότητες είναι κρίσιμες για το έργο μας. Αν κάποια κρίσιμη διαδικασία καθυστερήσει για μια μέρα, αυτό έχει σαν αποτέλεσμα να καθυστερήσει ολόκληρο το έργο κατά μια μέρα.
- Μπορούμε όμως να καθυστερήσουμε μη κρίσιμες διαδικασίες χωρίς να επηρεάζουμε το χρόνο ολοκλήρωσης. Το πιο πάνω είναι σωστό εάν η καθυστέρηση δεν αλλάζει το κρίσιμο μονοπάτι του έργου.

Υλοποιήσεις στο εργαστήριο



- Με τη βοήθεια του `lab12-template.c` θα υλοποιήσουμε τα ακόλουθα:
 - Topological sort
 - Τοπολογική αναφορά των γεγονότων ενός έργου
 - Earliest Event Time
 - Ο συντομότερος χρόνος στον οποίο μπορεί να συμβεί κάποιο γεγονός
 - Latest Event Time
 - Ο πιο αργός χρόνος στον οποίο μπορεί να συμβεί κάποιο γεγονός
 - Slack Time
 - Η δυνατή καθυστέρηση που μπορεί να έχει κάποια δραστηριότητα

Υλοποίηση topological sort



```
topSort1( graph G ){
    // αρχικοποίηση πίνακα μεγέθους |V|
    int I[|V|] = {};

    // μέτρηση in-degree για κάθε κόμβο
    for each vertex u
        for each edge (u,v)
            I[v]++;

    // προσπέλαση του γράφου
    for (i=1; i <= |V|; i++){
        v = FindVertexOfIndegree0;
        if (v == NULL) { // δεν υπάρχει κόμβος με in-degree=0
            Error("Graph has a cycle");
            return;
        }
        output v;           // εκτύπωση κόμβου
        for each edge (v,w) // μείωση in-degree
            I[w]--;          // για κάθε παιδί
    }
```

Υλοποίηση Earliest Event Time



- Έστω ότι $\text{earliestEventTime}(i)$ είναι ο πιο σύντομος χρόνος για τον οποίο το γεγονός i μπορεί να συμβεί. Το $\text{earliestEventTime}(i)$ είναι το μήκος του μεγαλύτερου κατευθυνόμενου μονοπατιού από την αρχή μέχρι το i . Για το i ισχύει το ακόλουθο:
 - $\text{earliestEventTime}(0) = 0$
 - $\text{earliestEventTime}(i) = \max \{ \text{earliestEventTime}(j) + \text{length}(j,i) \}, (j,i) \in E$,
 - E είναι το σύνολο των ακμών
 - $\text{length}(j,i)$ είναι ο χρόνος που χρειάζεται για την δραστηριότητα (j,i)
 - Το πρόγραμμα θα πρέπει να επιτρέπει ως είσοδο στην συνάρτηση οποιοδήποτε κόμβο του γράφου
 - Το earliestEventTime του τελικού γεγονότος ονομάζεται μήκος έργου (project length)

Υλοποίηση Latest Event Time



- Έστω ότι $\text{latestEventTime}(i)$ είναι ο πιο αργός χρόνος για τον οποίο το γεγονός i μπορεί να συμβεί. Εάν το γεγονός i δεν συμβεί μέχρι τον χρόνο αυτό, τότε το έργο δεν μπορεί να ολοκληρωθεί σε χρόνο ίσο με το μήκος έργου. Το latestEventTime του τελικού γεγονότος είναι ίσο με το earliestEventTime του τελικού γεγονότος. Για το i ισχύει το ακόλουθο:
 - $\text{latestEventTime}(n-1) = \text{earliestEventTime}(n-1)$
 - $\text{latestEventTime}(i) = \min \{ \text{latestEventTime}(j) - \text{length}(i,j) \}, (i,j) \in E,$
 - E είναι το σύνολο των ακμών
 - $\text{length}(j,i)$ είναι ο χρόνος που χρειάζεται για την δραστηριότητα (j,i)
 - Το πρόγραμμα θα πρέπει να επιτρέπει ως είσοδο στην συνάρτηση οποιοδήποτε κόμβο του γράφου

Υλοποίηση Slack Time



- Από τα προηγούμενα είδαμε ότι :
 - $\text{earliestEventTime}(i) = \max \{ \text{earliestEventTime}(j) + \text{length}(j,i) \}, (j,i) \in E,$
 - $\text{latestEventTime}(i) = \min \{ \text{latestEventTime}(j) - \text{length}(i,j) \}, (i,j) \in E.$
- Η δυνατή καθυστέρηση (slack time) $\text{slack}(i,j)$ της δραστηριότητας (i,j) ορίζεται ως:
 - $\text{slack}(i,j) = \text{latestEventTime}(i,j) - \text{earliestEventTime}(i,j)$
 - Μια δραστηριότητα ονομάζεται κρίσιμη εάν η δυνατή της καθυστέρηση είναι ίση με 0.
 - Αναγνωρίστε όλες τις κρίσιμες δραστηριότητες του παραδείγματος της διαφάνειας 3.