ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

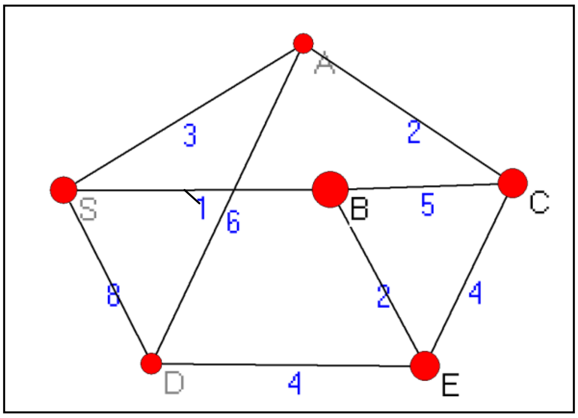
ΕΠΛ 035- ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΗΛΕΚΤΡΟΛΟΓΟΥΣ ΜΗΧΑΝΙΚΟΥΣ ΚΑΙ ΜΗΧΑΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΩΝ

Φροντιστηριακές Ασκήσεις - Γράφοι ΙΙ

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

1) Βραχύτερα Μονοπάτια σε Γράφους

Να δείξετε την εκτέλεση του αλγόριθμου του Dijkstra για εύρεση των ελάχιστων μονοπατιών από την κορυφή S προς όλες τις υπόλοιπες κορυφές του γράφου.



2) Τοπολογική Ταξινόμηση σε Γράφους

Ο Φαραώ της Αιγύπτου Alghamon IV σας έχει προσλάβει για να δουλέψετε στο κτίσιμο της πυραμίδας που θα χρησιμοποιηθεί για την ταφή του. Η πυραμίδα θα αποτελείται από n μεγάλες πέτρες $s_1,\ldots,s_n$. Οι αρχιτέκτονες της πυραμίδας έχουν υπολογίσει μία σχέση $\prec$ όπου, για δύο πέτρες s_i, s_j, ισχύει $s_i \prec s_j$ αν και μόνο αν η πέτρα s_i πρέπει να τοποθετηθεί πριν από την πέτρα s_j. Επίσης για κάθε πέτρα s γνωρίζουμε το χρόνο t(s) που απαιτείται για τοποθέτηση της στην σωστή θέση στην πυραμίδα.

Να σχεδιάσετε αποδοτικό αλγόριθμο ο οποίος να υπολογίζει τον ελάχιστο χρόνο που χρειάζεται για ολοκλήρωση της πυραμίδας, υποθέτοντας ότι πέτρες μπορούν να τοποθετηθούν στην θέση τους παράλληλα, εφόσον οι περιορισμοί της σχέσης $\prec$ δεν παραβιάζονται.

Φροντιστηριακές Ασκήσεις
Γράφοι ΙΙ - ΛΥΣΕΙΣ

Λύση Άσκηση 1

Πίνακας που δείχνει την εκτέλεση του Dijkstra (κόστος μετάβασης σε όλες τις κορυφές)

	D(S)	d(A)	d(B)	d(C)	d(D)	d(E)
∅	0	∞	∞	∞	∞	∞
{S}	0	3	1	∞	8	∞
{S,B}	0	3	1	6	8	3
{S,B,A}	0	3	1	5	8	3
{S,B,A,E}	0	3	1	5	7	3
{S,B,A,E,C}	0	3	1	5	7	3
{S,B,A,E,C,D}	0	3	1	5	7	3

dist(S,B)=1
+
dist(B,E)=2

	P(S)	P(A)	P(B)	P(C)	P(D)	P(E)
∅	-	-	-	-	-	-
{S}	-	S	S	-	S	-
{S,B}	-	S	S	B	S	B
{S,B,A}	-	S	S	A	S	B
{S,B,A,E}	-	S	S	A	E	B
{S,B,A,E,C}	-	S	S	A	E	B
{S,B,A,E,C,D}	-	S	S	A	E	B

Λύση Άσκηση 2

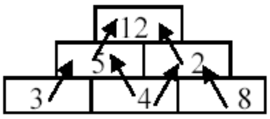
Το πρόβλημα μπορεί να μεταφραστεί σε πρόβλημα εύρεσης του μονοπατιού με το μέγιστο κόστος σε γράφο: Θεωρήστε τον γράφο με βάρη G=(V,E) όπου

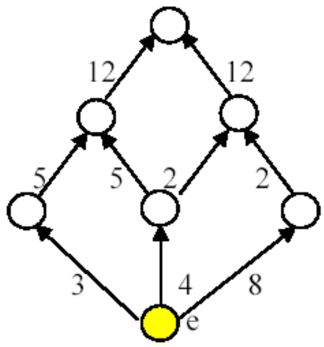
V={ $s_1,\ldots,s_n$ }, δηλαδή υπάρχει μία κορυφή για κάθε πέτρα και η επιπλέον κορυφή e (η γή),

E={(s_i, s_j)| αν $s_i \prec s_j$ }∪{(e, s_i) | αν δεν ισχύει $s_i \prec s_j$ για κανένα j}, και

βάρη w(u,v) = t(v).

Για παράδειγμα,





ο ελάχιστος χρόνος που χρειάζεται για ολοκλήρωση της πυραμίδας, (υποθέτοντας ότι πέτρες μπορούν να τοποθετηθούν στην θέση τους παράλληλα) είναι ίσος με το μέγιστο κόστος μονοπατιού από την κορυφή e στον πιο πάνω γράφο. Για να το υπολογίσουμε, μπορούμε να εφαρμόσουμε κάποια παραλλαγή της τοπολογικής ταξινόμησης.

```
LongestPath(vertex v, graph G){

    queue Q;
    int I[|V|];
   int mintime[|V|];

   για κάθε κορυφή u
        I[u]=0;
        mintime[u] = ∞;
   για κάθε κορυφή u
        για κάθε γείτονα v της u
             I[v]++;
   για κάθε κορυφή u
        if (I[u]==0)
             Enqueue(u, Q);
             mintime[u] = 0;
 while (! IsEmpty(Q)){
        u = Dequeue(Q);
        για κάθε γείτονα v της u
             I[v]--;
             if (I[v]==0) Enqueue (v, Q);
             mintime[v] = max(mintime[v],
                                    mintime[u]+ w(u,v));
```

// Το mintime[] κρατά για κάθε κόμβο τον ελάχιστο χρόνο διεκπεραίωσης

Αρχικοποίηση indegree table I[] και mintime[] κάθε κόμβου

Υπολογισμός indegree κάθε κόμβου

Προσθήκη όλων των κορυφών με 0 προαπαιτούμενα (i.e., e)

Ιδέα του Topological Sort όπου με χρήση BFS διασχίζουμε το γράφο.

Χρόνος εκτέλεσης: O(V+E)