ΓΕΩΡΓΙΟΣ ΙΩΑΝΝΙΔΗΣ ΑΕΜ : 10490
ΚΥΡΙΑΚΟΣ ΤΣΟΚΚΟΣ ΑΕΜ : 10496 **ΟΜΑΔΑ 19**
ΧΡΙΣΤΟΔΟΥΛΟΣ ΛΥΜΠΟΥΡΙΔΗΣ ΑΕΜ : 9525

ΠΡΟΒΛΗΜΑ 1ο

Υλοποίηση Αλγορίθμου:

```
FUNCTION (G=(V,E),w,s)
1.    for each vertex v ∈ G.V
2.         v.d = 0
3.         v.π = NIL
4.    end for
5.    s.d = 1
6.    S = NIL
7.    Q = G.V
8.      while Q != NULL
9.           u = extract-max (Q)
10.          S = S U {u}
11.          for each vertex ∈ G.Adj[u]
12.               RELAX(u,v,w)
13.          end for
14.    end while
RELAX(u,v,w)
1.  if ( u.d * w(u.v) > v.d  )
2.    v.d = u.d * w(u.v)
3.    v.π = u
```

Περιγραφή-Σχόλια:

Ο πάρα πάνω αλγόριθμος έχει ίδιο χρόνο εκτέλεσης με τον Dijkstra εφόσον είναι μία παραλλαγή του. Οι αλλαγές δεν είναι στους βρόγχους επανάληψης αλλά στις συνθήκες ελέγχου (if) και στις πράξεις οι οποίες δεν επηρεάζουν τον χρόνο εκτέλεσης.
Ο Αλγόριθμος αποτελείται από 2 συναρτήσεις. Αρχικά η FUNCTION δέχεται ως όρισμα έναν γράφο G το w και το s. Στο πρώτο for αρχικοποιώ κάθε κόμβο με (0,NIL). Μετά βάζω στον αρχικό κόμβο (s) την τιμή 1 που είναι η μέγιστη πιθανότητα . Έπειτα ορίζω το S κενό και το Q να περιέχει όλους τους κόμβους του γράφου. Στη συνέχεια εκτελείται ένας βρόχος while όσο το Q δεν είναι κενό. Βάζει στο u τον κόμβο με τη μεγαλύτερη πιθανότητα και τον αφαιρεί απο το Q. Μετά για κάθε γειτονικό κόμβο της u καλεί τη συνάρτηση RELAX.
Η RELAX έχει ορίσματα το u,v,w όπου v ο γειτονικός κόμβος και w η πιθανότητα να φτάσει από το u στο v. Αν η πιθανότητα του u πολλαπλασιασμένη με του w είναι μεγαλύτερη από του v ,αλλάζω την πιθανότητα του v με την πιθανότητα u*w και τον κόμβο που προέρχεται η v με u. Τέλος μετά την εκτέλεση του αλγορίθμου καταλήγω να έχω το μονοπάτι με την μεγαλύτερη πιθανότητα.

ΠΡΟΒΛΗΜΑ 2ο

Ερώτημα 1ο:
Υλοποίηση Αλγορίθμου:

```
FUNCTION (m,p,n,k)
1.      A[0]=0
2.      A[1]=p[1]
3.      for(i=2 to n)
4.           max=0
5.           for(j=1 to i -1)
6.                if( m[i] - m[j] > k  &&  A[j] + p[i] > max)
7.                   max= A[j]+p[j]
8.                end if
9.           end for
10.          A[i]=max
11.     end for
12.
13.   max2=A[1]
14.   for( i=2 to n )
15.      if( max2 < A[i] )
16.         max2 = A[i]
17.      end if
18.   end for
19.
20.  return max2
```

Τα υποπροβλήματα είναι ο υπολογισμός των μέγιστων προσδοκώμενων κερδών σε όλες τις τοποθεσίες.

Ερώτημα 2ο:
Περιγραφή-Σχόλια:
Αρχικά δημιουργούμε ένα πίνακα Α με n+1 στοιχεία, όπου n είναι ο αριθμός των πιθανών τοποθεσιών για να ανοίξει ένα εστιατόριο. Αρχικοποιούμε τον Α πίνακα με μηδέν σε όλες τις θέσεις όπου A[i] είναι το μέγιστο προσδοκώμενο συνολικό κέρδος που μπορούμε να πετύχουμε με το άνοιγμα ενός εστιατορίου στη τοποθεσία i . Αρχικοποιούμε την 1η θέση του πίνακα Α με 0 και τη δεύτερη θέση με το προσδοκώμενο κέρδος της πρώτης τοποθεσίας. Αυτό το κάνω αφού το προσδοκώμενο συνολικό κέρδος για καμία τοποθεσία είναι 0 και για τη πρώτη μόνο τοποθεσία θα είναι το κέρδος αυτής . Έπειτα υπολογίζουμε το μέγιστο κέρδος για κάθε τοποθεσία χρησιμοποιώντας διπλό βρόχο for.

Πιο συγκεκριμένα στο 1ο for εξετάζουμε κάθε τοποθεσία i από την 2η μέχρι και την τελευταία και στο 2ο for εξετάζω κάθε τοποθεσία προηγούμενη από την i-ωστή. Αρχικά στο 1ο for μηδενίζω μια μεταβλητή max. Στη συνέχεια έχω 1 if με δύο συνθήκες όπου η πρώτη ελέγχει τις αποστάσεις μεταξύ i-ωστής και j-ωστής τοποθεσίας , αν είναι μεγαλύτερη του k.Η δεύτερη συνθήκη ελέγχει αν το άθροισμα του κέρδους της i-ωστής τοποθεσίας με το μέγιστο προσδοκώμενο συνολικό κέρδος της j-ωστής τοποθεσίας είναι μεγαλύτερο του max. Αν ισχύουν και οι 2 συνθήκες αλλάζω το max με το άθροισμα που υπολόγισα. Όταν τελειώσει το 2ο for τοποθετώ το max στην θέση i του πίνακα Α. Με αυτόν τον τρόπο εξετάζω όλους τους πιθανούς συνδυασμούς τοποθεσιών για να ανοίξει ένα εστιατόριο και αν έχουμε το μέγιστο προσδοκώμενο κέρδος .Τέλος διατρέχω με ένα for τον πίνακα Α και επιστρέφω το μέγιστο στοιχείο που είναι το μέγιστο προσδοκώμενο συνολικό κέρδος.

Ερώτημα 3ο:

Ο πάρα πάνω αλγόριθμος έχει χρόνο εκτέλεσης $O(n^2)$ διότι χρησιμοποιούμε 3 βρόγχους for, 2 εμφωλευμένους και 1 μόνο του. Στους εμφωλευμένους έχω $O(n^2)$ ενω στον τελευταίο έχω $O(n)$. Οπότε $O(n^2)+O(n) = O(n^2)$.