

Algorithms - Assignment 2

Καζγκούτης Αθανάσιος

Παπαδόπουλος Δημήτριος-Λάζαρος

May 20, 2023

Πρόβλημα 1

Η συνάρτηση `MaxSuccessPath` παίρνει για ορίσματα τον γράφο $G = (V, E)$, το βάρος κάθε ακμής (p) το οποίο εκφράζει την πιθανότητα ότι ένα πακέτο το οποίο στέλνεται από τη μία συσκευή θα φτάσει στην άλλη χωρίς να χαθεί, τον αφετηριακό κόμβο s και τον κόμβο προορισμού t . Στην γραμμή 2 εκτελείται η απόδοση αρχικών τιμών. Στην γραμμή 3 ο αλγόριθμος δημιουργεί ένα κενό σύνολο (D) στο οποίο θα εισάγει τους κόμβους των οποίων οι τελικές πιθανότητες από την αφετηρία (s) έχουν ήδη προσδιοριστεί. Στην γραμμή 4 δημιουργούμε μία ουρά προτεραιότητας μεγίστου Q για τους κόμβους με κλειδιά τις τιμές των πεδίων p και ορίζουμε ως αρχικό περιεχόμενο το σύνολο των κόμβων. Στην

```
1 function MaxSuccessPath(G, p, s, t)
2   Initialize(G, s)
3   D = ∅
4   Q = G.V
5   while Q ≠ ∅
6     u = ExtractMaxQ
7     if (u == t)
8       PrintPath(u)
9     D = D ∪ {u}
10    for each vertex v ∈ G.Adj[u]
11      Relax(u, v, p)
```

```
1 function Initialize(G, s)
2   for each vertex v ∈ G.V
3     v.p = 0
4     v.π = NULL
5   s.p = 1
```

```
1 function Relax(u, v, p)
2   if (v.p < u.p · puv)
3     v.p = u.p · puv
4     v.π = u
```

Η διαδικασία της ΧΑΛΑΡΩΣΗΣ μιας ακμής (u, v) έχει ως εξής: ελέγχουμε εαν μπορούμε να βελτιώσουμε την μέγιστη πιθανότητα (p) της διαδρομής για τον κόμβο v διερχόμενοι μέσω του u . Η ποσότητα $v.p$ είναι η πιθανότητα λήψης του μηνύματος χωρίς λάθη. Η ποσότητα $v.π$ είναι ο προκατόχος του v . Ενώ το p_{uv} είναι η πιθανότητα σωστής λήψης μηνύματος ανάμεσα στους κόμβους u, v

```
1 function printPath(u)
2   i = 1
3   R[0] = u
4   while u.π ≠ NULL
5     R[i] = u.π
6     u = u.π
7     i = i + 1
8   for j = i-1 : -1 : 0
9     print → R[j]
```

Ο παραπάνω αλγόριθμος

Πρόβλημα 2