Ανάλυση και σχεδιασμός αλγορίθμων

Εργασία 2 - Ομάδα 58

Χατζηιωάννου Λαμπρινός, Ευαγγελίδης Νικόλαος, Φιλιππίδης Φοίβος-Παναγιώτης

2023-04-05

1 Πρόβλημα 1

1.1 Αλγόριθμος

```
αλγόριθμος Εύρεσης_Διαδρομής(ΓΡΑΦΟΣ, ΑΡΧΙΚΟΣ_ΚΟΜΒΟΣ, ΤΕΛΙΚΟΣ_ΚΟΜΒΟΣ)
  unvisited          = άδεια  σειρά προτεραιότητας, βάση πιθανότητας
  μονοπάτι           = άδεια λίστα

  για κάθε ΚΟΜΒΟΣ σε ΓΡΑΦΟΣ :
    ΚΟΜΒΟΣ.πιθανότητα = 0
    unvisited.add(ΚΟΜΒΟΣ)

  ΑΡΧΙΚΟΣ_ΚΟΜΒΟΣ.πιθανότητα = 1

  Όσο το unvisited δεν είναι άδειο:
    ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ = ΕΞΑΓΩΓΗ ΜΕΓΙΣΤΟΥ(unvisited)

    Για κάθε ΓΕΙΤΟΝΑΣ του ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ:
      # P(u,v) απο εκφώνηση η πιθανότητα να έχουμε επιτυχή μετάδοση επί της σύνδε
      # προσωρινή_πιθανότητα άφιξης στον ΓΕΙΤΟΝΑΣ
      προσωρινή_πιθανότητα = ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ.πιθανότητα * P(ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ, ΓΕΙΤΟΝ
      Αν προσωρινή_πιθανότητα > ΓΕΙΤΟΝΑΣ.πιθανότητα:
        ΓΕΙΤΟΝΑΣ.πιθανότητα = προσωρινή_πιθανότητα
        ΓΕΙΤΟΝΑΣ.γονέας = ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ

  ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ = ΤΕΛΙΚΟΣ_ΚΟΜΒΟΣ

  Όσο ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ.γονέας δεν είναι κενό:
    μονοπάτι.prepend(ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ)
    ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ = ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ.γονέας
```

```
μονοπάτι.prepend(ΑΡΧΙΚΟΣ_ΚΟΜΒΟΣ)

Επίστρεψε μονοπάτι
```

1.2 Περιγραφή του αλγορίθμου

Για να μπορέσουμε να βρούμε την διαδρομή μέγιστης πιθανότητας από μια συσκευή στην άλλη χρησιμοποιήσαμε μια δική μας παραλλαγή του αλγορίθμου του Dijkstra. Αξιοποιούμε τις ιδιότητες των κόμβων (πιθανότητα και γονέας), να αποθηκεύουμε την μέγιστη πιθανότητα επιτυχούς αποστολής (από τον `ΑΡΧΙΚΟ_ΚΟΜΒΟ,` μέχρι τον εξεταζόμενο) αλλά και τον προηγούμενο κόμβο μέσω του οποίου φτάσαμε στον επιθυμητό κόμβο, αντίστοιχα. Εφόσον αρχικοποιήσουμε την πιθανότητα για κάθε κόμβο σε τιμή 0 και τους προσθέσουμε στην ουρά προτεραιότητας unvisited (στην οποία τα στοιχεία βρίσκονται ταξινομημένα βάση της ιδιότητας πιθανότητα που αναφέρθηκε παραπάνω), τους εξάγουμε από την σειρά έναν έναν, βάση της μέγιστης πιθανότητας. Στην συνέχεια, μέχρι να αδειάσει η unvisited, ελέγχουμε κάθε `ΓΕΙΤΟΝΑ` του κάθε `ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ,` για την περίπτωση που η συντομότερη διαδρομή στον `ΓΕΙΤΟΝΑ` από τον `ΑΡΧΙΚΟ_ΚΟΜΒΟ,` διέρχεται από τον `ΤΩΡΙΝΟΣ_ΚΟΜΒΟΣ.` Σε αυτή την περίπτωση παρατηρείται πως η `προσωρινή_πιθανότητα` είναι μεγαλύτερη από την (ένθετη ιδιότητα) πιθανότητα του `ΓΕΙΤΟΝΑ,` οπότε έχουμε ενημέρωση των πεδίων πιθανότητα και γονέας του `ΓΕΙΤΟΝΑ.` Τότε, ξεκινώντας από την `ΤΕΛΙΚΟΣ_ΚΟΜΒΟΣ` (στην δικιά μας περίπτωση από την συσκευή t) ακολουθάμε την ιδιότητα γονέας μέχρι να φτάσουμε στην `ΑΡΧΙΚΟΣ_ΚΟΜΒΟΣ` (στην δικιά μας περίπτωση στην συσκευή s) και προσθέτουμε έναν έναν τους κόμβους στην αρχή της λίστας "μονοπάτι" μέσω της μονοπάτι.prepend. Τελικά, επιστρέφεται η λίστα "μονοπάτι" η οποία μας δείχνει την διαδρομή που πρέπει να ακολουθήσουμε για να πετύχουμε την μέγιστη πιθανότητα

1.3 Χρονική ανάλυση

Ο αλγόριθμος μας έχει ίδια δομή με του Dijkstra, ωστόσο έχει κάποιες διαφοροποιήσεις ώστε να επιλέγεται η διαδρομή μεγαλύτερου βάρους (μεγαλύτερης πιθανότητας) και τα βάρη αντί να προσθέτονται να πολλαπλασιάζονται μεταξύ τους.

Αν V είναι ο αριθμός των κόμβων του γράφου, τότε:

- Η πρώτη επανάληψη του αλγορίθμου παίρνει $O(V)$ χρόνο καθώς αρχικοποιείται από μια φορά ο κάθε κόμβος
- Η δεύτερη επανάληψη του αλγορίθμου εκτελείται V φορές, καθώς ένα προς ένα ελέγχει και αφαιρεί κόμβους από την λίστα unvisited. Μέσα σε αυτή την επανάληψη υπάρχει μια ακόμα η οποία τρέχει το πολύ V φορές και εξαρτάται από τους πόσους γείτονες έχει ο κόμβος

Οπότε ο χρόνος εκτέλεσης του κώδικα είναι:

$$O(V) + O(V) * O(V) = O(V^2) \tag{1}$$

Ίδιος με του αλγοριθμού του Dijkstra

2 Πρόβλημα 2

2.1 Ανάλυση προβλήματος

Έχοντας ως δεδομένο τις πιθανές τοποθεσίες από 1 εώς i, πρέπει να βρεθεί το μέγιστο συνολικό προσδοκόμενο κέρδος, τοποθετώντας τα εστιατόρια στις τοποθεσίες, έτσι ώστε η ελάχιστη απόσταση να είναι τουλάχιστον k μέτρα.

Ορίζουμε το μέγιστο συνολικό προσδοκόμενο κέρδος των πρώτων i εστιατορίων, ή τοποθεσιών καλύτερα ως $Q(i)$. Συνεπώς ορίζουμε την αναδρομική σχέση ως

$$Q(i) = max\,\{Q(i-1), Q(l_s) + p[i]\} \tag{2}$$

Όπου: $l_s < i$ ο μέγιστος ακέραιος για τον οποίο ικανοποιείται η συνθήκη εγγύτητας:

$$m[i] - m[l_s] \geq k \tag{3}$$

Ορίζεται δηλαδή ως το μέγιστο μεταξύ του μέγιστου συνολικού προσδοκόμενου κέρδους όλων των προηγούμενων μαγαζιών, και του μέγιστου συνολικού προσδοκόμενου κέρδους του πρώτου στοιχείου που τηρεί την (**??**), αυξημένου κατά το νέο κέρδος $p[i]$

Όπως φαίνεται, για να λύσουμε το πρόβλημα μεγέθους n το ανάγουμε σε υποπρόβλημα μεγέθους $n-1$.

2.2 Αλγόριθμος

Η σχέση που μοντελοποιήσαμε παραπάνω εύκολα υλοποιείται με αναβιβαστική εκδοχή δυναμικού προγραμματισμού ως εξής:

```
μέγιστο_προσδοκόμενο_κέρδος (n,m,p,k):
    # Αρχικοποίηση της λίστας μέγιστου κέρδους
    q = p

    # Κάθε στιγμή, για την i-οστή τοποθεσία:
    # Το p[i] : κέρδος
    # Το m[i] : απόσταση από την αρχή της Εγνατίας
    # Το q[i] : μέγιστο συνολικό προσδοκόμενο κέρδος μέχρι εκεί

    # Η προηγούμενη τοποθεσία που ικανοποιεί την συνθήκη εγγύτητας
    ΠΡΟΗΓΤΟΠΟΘΕΣΙΑ = 1;

    Για κάθε ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ στο [2,n]:
        Για κάθε ΤΟΠΟΘΕΣΙΑ ΣΤΟ [ΠΡΟΗΓΤΟΠΟΘΕΣΙΑ, ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ]:
            Αν m[ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ] - m[ΤΟΠΟΘΕΣΙΑ] < k:
                ΠΡΟΗΓΤΟΠΟΘΕΣΙΑ = ΤΟΠΟΘΕΣΙΑ - 1
                q[ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ] = max(q[ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ-1],
                            q[ΠΡΟΗΓΤΟΠΟΘΕΣΙΑ] + p[ΤΩΡΙΝΗΤΟΠΟΘΕΣΙΑ])
```

```
                break;

    # Αφότου τελειώσουν και οι δύο επαναλήψεις
    Επίστρεψε q[n]
```

2.3 Απόδειξη ορθότητας

Θα αποδείξουμε την ορθότητα του αλγόριθμου βάση του επαγωγικού συλλογισμού, για $n \geq 1$.

Στην αρχική περίπτωση όπου $n = 1$, μόνο μία θέση για *άνοιγμα* εστιατορίου υπάρχει, και το κέρδος της είναι η επιστρεφόμενη τιμή.

Τώρα θα αποδείξουμε ότι αν ο αλγόριθμος επιστρέφει το σωστό μέγιστο συνολικό προσδοκόμενο κέρδος για τα πρώτα $l - 1$ στοιχεία, τότε θα επιστρέφει το σωστό προσδοκόμενο κέρδος και για τα πρώτα l στοιχεία.

Αρχικά, κατά την εκτέλεση του αλγοριθμου ο πίνακας q σταδιακά διαμορφώνεται σε αύξουσα σειρά. Δηλαδή, όταν εξετάζεται το l-οστό σημείο ισχύει:

$$\forall a, b < l : a < b \iff q[a] \leq q[b] \tag{4}$$

Έστω l_s, το πρώτο (εκ του τέλους) στοιχείο του m, που απέχει τουλάχιστον k , από το l. Τότε, το μέγιστο συνολικό προσδοκόμενο κέρδος των πρώτων l στοιχείων θα είναι το μέγιστο μεταξύ των παρακάτω:

- Το ήδη γνωστό μέγιστο συνολικό προσδοκόμενο κέρδος των πρώτων $l - 1$ στοιχείων
- Την ποσότητα $q[l_s] + p[l]$ η οποία ουσιαστικά αποτελεί το μέγιστο συνολικό προσδοκόμενο κέρδος του πλήθους στοιχείων που επιτρέπουν την κατασκευή του εστιατορίου στην θέση l, αυξημένη κατά τα έσοδα του μαγαζιού.

2.4 Χρονική ανάλυση

Ο αλγόριθμος μας αποτελείται από 2 nested loops, η πολυπλοκότητα της κάθε μίας είναι της τάξης $O(n)$. Ο αλγόριθμος μας, όμως, δεν τρέχει με πολυπλοκότητα $O(n^2)$, καθώς η εσωτερική επανάληψη δεν θα τρέξει παρα μόνο n φορές, σε όλες τις επαναλήψεις του εξωτερικού βρόχου. Ως εκ τούτου ο αλγόριθμός μας τρέχει με γραμμική πολυπλοκότητα $O(n)$.