Αλγόριθμοι και Πολυπλοκότητα

3η σειρά προγραμματιστικών ασκήσεων

Σχέδιο Λύσεων

CoReLab

ΣΗΜΜΥ Ε.Μ.Π.

17 Φεβρουαρίου 2020

Outline

Προγραμματιστική Άσκηση 1: Αποστάσεις σε δίκτυο με εκθετικά Μήκη Ακμών

Είσοδος: N πόλεις που συνδέονται με M μη κατευθυνόμενες ακμές, κάθε μία με διαφορετικό βάρος δύναμη του 2 (μικρότερο βάρος 2^0 και μεγαλύτερο $2^{|M|-1}$).

Έξοδος: Δυαδική αναπαράσταση του αθροίσματος των ελάχιστων διαδρομών μεταξύ όλων των ζευγών των πόλεων.

Προγραμματιστική Άσκηση 1: Αποστάσεις σε δίκτυο με εκθετικά Μήκη Ακμών

Ξεκινάμε με την ακόλουθη παρατήρηση: Κάθε μονοπάτι μεταξύ δυο κόμβων v, u του MST ενός γράφου G της συγκεκριμένης μορφής είναι συντομότερο μονοπάτι μεταξύ των v και u στον G.

Απόδειξη: Όπως έχουμε ήδη δείξει (βλ. σειρά 3, ασκ. 6 γραπτών ασκήσεων), το μονοπάτι, $p_{v,u}$ που συνδέει τους v και u στο MST περιέχει την bottleneck ακμή, $eb_{v,u}$, (ελάχιστη μέγιστη ακμή μεταξύ όλων των μονοπατιών) μεταξύ των v και u. Έστω οποιοδήποτε άλλο μονοπάτι $p'_{v,u}$ μεταξύ των v και u που είναι ίδιο από τον v μέχρι τον v' και από τον u' μέχρι τον u για $v', u' \in p_{v,u}$ και διαφέρει στο υπομονοπάτι $p_{v',u'}$. Το $p'_{v',u'}$ έχει τουλάχιστον μια ακμή e' μεγαλύτερου η ίσου βάρους της $eb_{v',u'}$. Όμως, σε ένα γράφο της μορφής του προβλήματος αυτό σημαίνει ότι $w_{e'} > 2w_{eb_{v',u'}}$, ενώ ξέρουμε ότι κάθε ακμή στο $p_{v',u'}$ είναι διαφορετική δύναμη του 2, με μεγαλύτερη την $eb_{v',u'}$ συνεπώς $w_{p_{v',u'}} < 2w_{eb_{v',u'}}$. Άρα και $p_{v,u} < p'_{v,u}$.

Προγραμματιστική Άσκηση 1: Αποστάσεις σε δίκτυο με εκθετικά Μήκη Ακμών

Δεύτερη παρατήρηση: Αν έχουμε το MST του γράφου, μπορούμε τρέχοντας ένα DFS ξεκινώντας από ένα φύλλο να δούμε πόσες φορές περνάμε από κάθε ακμή (δηλαδή, σε πόσα ελάχιστα μονοπάτια συμμετέχει η κάθε ακμή).

Τρίτη παρατήρηση: Έστω ότι ξέρουμε ότι η ακμή με βάρος 2^{c_k} εμφανίζεται x_k φορές. Ξέρουμε ότι η συγκεκριμένη ακμή επηρεάζει μόνο από το $c_k + 1$-οστό least significant bit και πάνω. Επιπλέον, μπορεί να επηρεάσει το πολύ $log(x_k)$ bits. Συνεπώς αν ξεκινήσουμε να φτιάχνουμε την συμβολοσειρά εξόδου προσθέτοντας αρχικά το μικρότερο βάρος και έπειτα τα μεγαλύτερα, θα έχουμε το πολύ $O(M)$ προσθέσεις που αφορούν το πολύ σε $log(max(x_k))$ bits.

Προγραμματιστική Άσκηση 1: Αποστάσεις σε δίκτυο με εκθετικά Μήκη Ακμών

Έτσι κατασκευάζουμε τον εξής αλγόριθμο:

- Βρες ένα *MST* του γράφου με Kruskal. Δείτε ότι οι ακμές μπορούν να ταξινομηθούν εύκολα με σουντινγ σορτ.
- Τρέξε ένα *DFS* στο παραπάνω *MST* και κατέγραψε το πλήθος εμφανίσεων κάθε ακμής.
- Οι ακμές προστίθενται στο *MST* ταξινομημένες, με βάση το βάρος. Με βάση αυτό, φτιάξε την έξοδο προσθέτωντας από τη μικρότερη (επί πλήθος εμφανίσεων) στην μεγαλύτερη, κοιτώντας κάθε φορά μόνο το εύρος των bits που επηρεάζονται

Προσέξτε ότι όταν τελειώσουμε με την ακμή βάρους 2^{c_k} μπορούμε να έχουμε εμφανίσει τα πρώτα $c_k + 1$ bits (και χρειάζεται να κρατάμε έναν buffer μήκους το πολύ $log(max(x_k))$ πριν κανούμε τις υπόλοιπες προσθέσεις.

Προγραμματιστική Άσκηση 1: Αποστάσεις σε δίκτυο με εκθετικά Μήκη Ακμών

Κάθε βήμα έχει την εξής πολυπλοκότητα:

- $O(MlogN)$ για MST.
- $O(M)$ για DFS.
- $O(Mlog(max(x_k)))$ για πρόσθεση. Αλλά $max(x_k) < M^2$ (Γιατί τα ελάχιστα μονοπάτια είναι το πολύ όσο όλα τα πιθανά ζεύγη κόμβων και $N \leq E$).

Έτσι, συνολικά:

Πολυπλοκότητα

$$\mathcal{O}(MlogM)$$

Outline

Είσοδος: Οι αριθμοί N, M, K και X και οι συντεταγμένες των πιλοτικών διαβάσεων.

Έξοδος: Το πλήθος (mod 1000000103) όλων των μονοπατιών, που χρησιμοποιούν το πολύ X πιλοτικές διαβάσεις και οδηγούν το υποβρύχιο από την αρχική του θέση, με συντεταγμένες (N - 1, M - 1), στην τελική του θέση, με συντεταγμένες (0, 0).

Λύση1:

- Θεωρούμε τον τρισδιάστατο πίνακα $grid[N][M][X+1]$, όπου το στοιχείο $grid[i][j][k]$ εκφράζει το πλήθος (mod 1000000103) όλων των μονοπατιών που χρησιμοποιούν ακριβώς k πιλοτικές διαβάσεις και οδηγούν το υποβρύχιο από την θέση, με συντεταγμένες (i, j), στην τελική του θέση, με συντεταγμένες $(0, 0)$.
  - Ισχύει ότι $grid[0][0][0] = 1$ και $grid[0][0][k] = 0$, για κάθε k ε $(0, X + 1)$.
  - Στην περίπτωση, που δεν ξεκινά κάποια πιλοτική διάβαση από το σημείο (i, j), ισχύει ότι $grid[i][j][k] = (grid[i][j - 1][k] + grid[i - 1][j][k])$ mod 1000000103 (ειδικές περιπτώσεις για $i = 0$ ή $j = 0$), για κάθε $k < X + 1$.
  - Έστω, ότι από το σημείο (i, j) ξεκινά πιλοτική διάβαση, η οποία καταλήγει στο σημείο (x, y), τότε ισχύει ότι $grid[i][j][0] = 0$ και $grid[i][j][k] = grid[x][y][k - 1]$, για κάθε k ε $(0, x + 1)$.

Προγραμματιστική Άσκηση 2: Υποβρύχιο Τρίτων

- Σκοπός μας είναι να υπολογίσουμε το $(\sum_{k=0}^{X} grid[N - 1][M - 1][k])\ mod\ 1000000103$.
- Ξεκινώντας από το σημείο (0, 0), μπορούμε είτε να εφαρμόσουμε *BFS* στο πλέγμα, είτε να εκμεταλλευτούμε το γεγονός ότι η κλίση κάθε πιλοτικής διάβασης ως προς τον οριζόντιο άξονα είναι τουλάχιστον 90 μοίρες και να διασχίσουμε το πλέγμα από αριστερά προς τα δεξιά και από πάνω προς τα κάτω, για ευκολία και εξοικονόμηση χώρου.

Πολυπλοκότητα:

- Χρονική, $O(NMX)$.
- Χωρική, $O(MX)$.

Λύση 2:

- Αρχικά κατασκευάζουμε ένα κατευθυνόμενο δέντρο T, με βάρη στις ακμές. Το T, αποτελείται από συνολικά $K + 2$ κορυφές, μία για κάθε πιλοτική διάβαση και άλλες δύο για το αρχικό και το τελικό σημείο του πλέγματος, αντίστοιχα. Μεταξύ δύο κορυφών A και B, του T, υπάρχει ακμή με κατεύθυνση από το A, προς το B, ανν υπάρχει μονοπάτι, που να συνδέει το τελικό σημείο της πιλοτικής διάβασης, που αντιστοιχεί στην κορυφή B, με το αρχικό σημείο της πιλοτικής διάβασης, που αντιστοιχεί στην κορυφή A. Επίσης, υπάρχει μια ακμή από την κορυφή, που αντιστοιχεί στο τελικό σημείο του πλέγματος, προς όλες τις άλλες κορυφές (αντιμετωπίζεται σαν αρχικό σημείο πιλοτικής διάβασης) και μια ακμή από κάθε κορυφή του T, προς την κορυφή που αντιστοιχεί στο αρχικό σημείο του πλέγματος (αντιμετωπίζεται σαν τελικό σημείο πιλοτικής διάβασης).

- Το βάρος κάθε ακμής, αποτελείται από ένα διάνυσμα, που περιέχει τα πλήθη όλων των μονοπατιών που συνδέουν τα δύο σημεία χρησιμοποιώντας ακριβώς $k < X+1$ πιλοτικές διαβάσεις.
- Στην συνέχεια, εφαρμόζουμε ένα *BFS* στο Τ και υπολογίζουμε το ζητούμενο πλήθος μονοπατιών.
- Για να αποφύγουμε υπερχείλιση στα ενδιάμεσα αποτελέσματα, θα χρειαστεί να κάνουμε χρήση των ιδιοτήτων της πράξης *mod*, καθώς και του αλγορίθμου ύψωσης σε δύναμη.

Πολυπλοκότητα:

- Στην ουσία, θα χρειαστεί να υπολογίσουμε το πλήθος των μονοπατιών, μεταξύ όλων των αρχικών σημείων των πιλοτικών διαβάσεων, αλλά και όσων συνδέουν την αρχή μιας πιλοτικής διάβασης με το τέλος μιας άλλης για όλα τα $k < X$. Πολυπλοκότητα $O(X * K^2)$.
- Το *BFS* στο T έχει επίσης χρονική πολυπλοκότητα $O(X * K^2)$.
- Άρα συνολικά $O(X * K^2)$ ή απλά $O(K^3)$.
- Δηλαδή, ο συγκεκριμένος αλγόριθμος λύνει στιγμιότυπα με αυθαίρετα μεγάλα N και M.