Αλγόριθμοι και Πολυπλοκότητα

3η Σειρά Ασκήσεων

Σχέδιο Λύσεων

CoReLab

ΣΗΜΜΥ Ε.Μ.Π.

24 Ιανουαρίου 2020

Είσοδος: m κλειδιά, n κουτιά και οδηγίες για το ποιο κλειδί ανοίγει ποιο κουτί

Έξοδος: Ποια κουτιά πρέπει να σπάσουμε και με ποια σειρά πρέπει να ξεκλειδώσουμε τα υπόλοιπα για να ανακτήσουμε όλα τα κλειδιά σπάζοντας τα λιγότερα κουτιά.

Λύση: Μπορούμε να αντιστοιχίσουμε τα στοιχεία της εισόδου σε ένα κατευθυνόμενο γράφημα ως εξής:

- κουτί i που περιέχει κλειδιά $\rightarrow$ κορυφή i
- κλειδί που βρίσκεται στο κουτί i και ξεκλειδώνει το κουτί j $\rightarrow$ ακμή (i, j)

Αγνοούμε τις ακμές τύπου (i, i) γιατί δεν μας είναι χρήσιμες. Το γράφημα έχει το πολύ n κορυφές και το πολύ m ακμές.

Άσκηση 1 - Ο Άγιος Βασίλης και τα Χρυσά Κλειδιά

1. Υπολογίζουμε το γράφημα Ισχυρά Συνεκτικών Συνιστωσών.
2. Βρίσκουμε τις ΙΣΣ που είναι πηγές.
3. Για κάθε ΙΣΣ που είναι πηγή
   - σπάμε ένα κουτί που ανήκει σε αυτή
   - ξεκινώντας από αυτό το κουτί ξεκλειδώνουμε τα υπόλοιπα κουτιά στα οποία έχουμε πρόσβαση τρέχοντας DFS

Ορθότητα: Ο αλγόριθμος σπάει ένα κουτί σε κάθε Ισχυρά Συνεκτική Συνιστώσα που είναι πηγή. Αυτή η κίνηση είναι απαραίτητη καθώς δεν μπορούμε να ξεκλειδώσουμε κανένα κουτί μιας ΙΣΣ πηγής αν όλα τα κουτιά που ανήκουν σε αυτή είναι κλειστά. Επίσης δε χρειάζεται να σπάσουμε κανένα επιπλέον κουτί γιατί το γράφημα ΙΣΣ μας εξασφαλίζει ότι τα υπόλοιπα κουτιά μπορούν να ξεκλειδωθούν.

Άσκηση 1 - Ο Άγιος Βασίλης και τα Χρυσά Κλειδιά

Πολυπλοκότητα: Χρησιμοποιώντας τον αλγόριθμο Kosaraju υπολογίζουμε το γράφημα ΙΣΣ σε $O(|V| + |E|)$. Η εύρεση των ΙΣΣ του γραφήματος που είναι πηγές έχει πολυπλοκότητα $O(|V'| + |E'|)$, όπου V' οι ΙΣΣ και E' οι ακμές του γραφήματος ΙΣΣ. Το τελικό DFS έχει πολυπλοκότητα $O(|V| + |E|)$.

Επομένως συνολικά ο αλγόριθμος έχει υπολογιστική πολυπλοκότητα $O(n + m)$.

Άσκηση 2 - Μακρύτερο Μονοπάτι σε Δέντρο

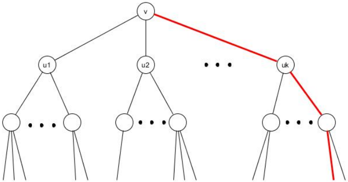
Είσοδος: Δέντρο $T(V, E)$ με βάρη $w(u) \in \mathbb{Z}$, για κάθε κόμβο $u \in V$ (πιθανώς αρνητικά).

Ζητείται: Απλό μονοπάτι p που να μεγιστοποιεί το συνολικό βάρος $w(p) = \sum_{u \in p} w(u)$.

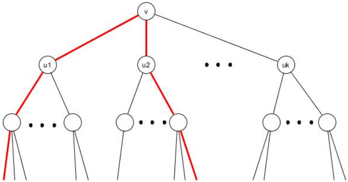
Άσκηση 2: Μακρύτερο Μονοπάτι σε Δέντρο

Ιδέα: Έστω ότι επιλέγουμε κόμβο v και τον θεωρούμε ρίζα του δέντρου, με παιδιά τους κόμβους $u_1, \ldots, u_k$. Τότε έχουμε τις εξής δυο βασικές περιπτώσεις:

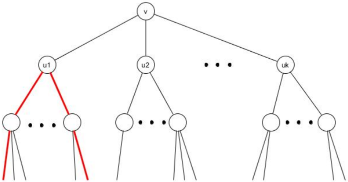
- Π1. Ο v ανήκει στο μονοπάτι p: Διακρίνουμε δυο υποπεριπτώσεις:
  - Π1.α. Ο v είναι "ακριανός" κόμβος του μονοπατιού: Σε αυτήν την περίπτωση, το μονοπάτι p, αν αφαιρέσουμε τον v, είναι το μέγιστου βάρους μονοπάτι που βρίσκεται σε ένα από τα υποδέντρα με ρίζα $u_1, \ldots, u_k$ και "καταλήγει" στην αντίστοιχη ρίζα.
  - Π1.β. Ο v είναι "ενδιάμεσος" κόμβος του μονοπατιού: Σε αυτήν την περίπτωση, αν αφαιρέσουμε τον v, το p "σπάει" σε δυο μονοπάτια, τα δυο μέγιστου βάρους που βρίσκονται στα υποδέντρα με ρίζα $u_1, \ldots, u_k$ και "καταλήγουν" στην αντίστοιχη ρίζα (σε διαφορετικά υποδέντρα).
- Π2. Ο v δεν ανήκει στο p: Τότε, το μέγιστο μονοπάτι βρίσκεται σε κάποιο από τα υποδέντρα με ρίζα $u_1, \ldots, u_k$.



(α′) Π1.α



(β′) Π1.β



(γ′) Π2

Άσκηση 2: Μακρύτερο Μονοπάτι σε Δέντρο

Λύση: Για κάθε κόμβο u κρατάμε δυο τιμές, $w_e(u)$ και $w_m(u)$. Στο υποδέντρο με ρίζα τον u, το $w_e(u)$ συμβολίζει το μέγιστο βάρος μονοπατιού που "καταλήγει" στον u και το $w_m(u)$ συμβολίζει το μέγιστο βάρος μονοπατιού που περιέχει ως ενδιάμεσο τον u.
Έτσι, αν γνωρίζουμε τα $w_e(u_1)$, ..., $w_e(u_k)$, για να υπολογίσουμε τα $w_e(v)$ και $w_m(v)$ έχουμε τις εξής (αναδρομικές) σχέσεις:

1. $w_e(v) = w(v) + \max\{\max_{u \in \{u_1,\ldots,u_k\}}\{w_e(u)\}, 0\}$ (*).
2. $w_m(v) =$
   $w(v) + \max\{\max_{u_i,u_j \in \{u_1,\ldots,u_k\} | u_i \neq u_j}\{w_e(u_i) + w_e(u_j)\}, 0\}$

- Παρατηρήστε ότι αρκεί να θυμόμαστε τα δυο μέγιστα $w_e(u_i), w_e(u_j) \in \{w_e(u_1), \ldots, w_e(u_k)\}$.
- Τρέχουμε αναδρομικά για κάθε κόμβο του δέντρου, με *DFS*.
- Για κάθε κόμβο θυμόμαστε ποιό (η ποιά - για το $w_m(u)$) είναι το παιδί του επιλέξαμε.

Άσκηση 2: Μακρύτερο Μονοπάτι σε Δέντρο

Λύση: Προσέχουμε (*) ότι αν όλα τα τα μέγιστα μονοπάτια που καταλήγουν στα παιδιά του υπό αναβάθμιση κόμβου u είναι αρνητικά, κρατάμε μόνο το βάρος του κόμβου - και θυμόμαστε ότι δεν επιλέξαμε κανένα παιδί του (δηλ. αν ο u ανήκει στην τελική λύση, θα είναι “ακριανός” κόμβος). Έτσι “αποκόπτουμε” τα αρνητικά “υπομονοπάτια”.

- Τέλος, σε κάθε αναβάθμιση κόμβου θυμόμαστε ποια είναι τα μέγιστα w_e και w_m (w_{e_max} και w_{m_max}) που έχουμε συναντήσει.
- Ξεκινώντας από τον κόμβο με το μέγιστο εκ των w_{e_max} και w_{m_max} και ακολουθώντας αναδρομικά τα επιλεγμένα παιδιά, έχουμε το ζητούμενο μονοπάτι.
- Παρατηρήστε, ότι αν δεν υπάρχει κανένα “θετικό” μονοπάτι με τουλάχιστον μια ακμή, το μέγιστο μονοπάτι είναι ο κόμβος με το μεγαλύτερο βάρος.

Πολυπλοκότητα:

- Όπως είδαμε, αν για κόμβο v με παιδιά $u_1, \ldots, u_k$ θυμόμαστε τα δύο μέγιστα $w_e(u_i)$, $w_e(u_j)$ κατά την αναδρομή, τα $w_e(v)$ και $w_m(v)$ υπολογίζονται σε $O(1)$.
- Οπότε κρατάμε δυο τιμές, $mx_1(v)$, $mx_2(v)$ για κάθε κόμβο v, που αντιστοιχούν στα δυο μέγιστα $w_e(u_i), w_e(u_j) \in \{w_e(u_1), \ldots, w_e(u_2)\}$ και τις ενημερώνουμε κάθε φορά που υπολογίζουμε ένα $w_e(u_l) \in \{w_e(u_1), \ldots, w_e(u_2)\}$.
- Η σειρά του *DFS* εξασφαλίζει ότι θα έχουμε υπολογίσει τα $mx_1(v)$, $mx_2(v)$ όταν φτάσουμε στον υπολογισμό των τιμών του v.

Πολυπλοκότητα

$$O(|V|)$$

Άσκηση 3: Κλέφτες και Αστυνόμοι

Είσοδος: Μη κατευθυνόμενος γράφος κινήσεων

Έξοδος: Νικητής του παιχνιδιού για κάθε συνδυασμό αρχικών θέσεων

Λύση (Ιδέα):

- Δημιουργούμε ένα νέο γράφο με $O(V^2)$ κόμβους και $O(VE)$ ακμές όπου ο κάθε κόμβος είναι της μορφής (θέση Κώστα, θέση Ανδρέα, επόμενη κίνηση) και κάθε ακμή της μορφής ((K,A,K), (K',A,A)) ή ((K,A,A), (K,A',K)), όπου οι (K,K') και (A,A') είναι ακμές του αρχικού γράφου
- Θα δώσουμε σε κάθε κόμβο μια ετικέτα A, K ή I αν ξεκινώντας απ' αυτή την κατάσταση το παιχνίδι καταλήγει σε νίκη του Ανδρέα, του Κώστα ή έρχεται ισόπαλο
- Αν για ένα κόμβο γνωρίζουμε την ετικέτα όλων των γειτόνων του μπορούμε εύκολα να βρούμε τη δική του

Άσκηση 3: Κλέφτες και Αστυνόμοι

Είσοδος: Μη κατευθυνόμενος γράφος κινήσεων

Έξοδος: Νικητής του παιχνιδιού για κάθε συνδυασμό αρχικών θέσεων

Λύση (Ιδέα):

- Δημιουργούμε ένα νέο γράφο με $O(V^2)$ κόμβους και $O(VE)$ ακμές όπου ο κάθε κόμβος είναι της μορφής (θέση Κώστα, θέση Ανδρέα, επόμενη κίνηση) και κάθε ακμή της μορφής ((K,A,K), (K',A,A)) ή ((K,A,A), (K,A',K)), όπου οι (K,K') και (A,A') είναι ακμές του αρχικού γράφου
- Θα δώσουμε σε κάθε κόμβο μια ετικέτα A, K ή I αν ξεκινώντας απ' αυτή την κατάσταση το παιχνίδι καταλήγει σε νίκη του Ανδρέα, του Κώστα ή έρχεται ισόπαλο
- Αν για ένα κόμβο γνωρίζουμε την ετικέτα όλων των γειτόνων του μπορούμε εύκολα να βρούμε τη δική του

Άσκηση 3: Κλέφτες και Αστυνόμοι

Είσοδος: Μη κατευθυνόμενος γράφος κινήσεων

Έξοδος: Νικητής του παιχνιδιού για κάθε συνδυασμό αρχικών θέσεων

Λύση (Ιδέα):

- Δημιουργούμε ένα νέο γράφο με $O(V^2)$ κόμβους και $O(VE)$ ακμές όπου ο κάθε κόμβος είναι της μορφής (θέση Κώστα, θέση Ανδρέα, επόμενη κίνηση) και κάθε ακμή της μορφής ((K,A,K), (K',A,A)) ή ((K,A,A), (K,A',K)), όπου οι (K,K') και (A,A') είναι ακμές του αρχικού γράφου
- Θα δώσουμε σε κάθε κόμβο μια ετικέτα A, K ή I αν ξεκινώντας απ' αυτή την κατάσταση το παιχνίδι καταλήγει σε νίκη του Ανδρέα, του Κώστα ή έρχεται ισόπαλο
- Αν για ένα κόμβο γνωρίζουμε την ετικέτα όλων των γειτόνων του μπορούμε εύκολα να βρούμε τη δική του

Άσκηση 3: Κλέφτες και Αστυνόμοι

Είσοδος: Μη κατευθυνόμενος γράφος κινήσεων

Έξοδος: Νικητής του παιχνιδιού για κάθε συνδυασμό αρχικών θέσεων

Λύση (Ιδέα):

- Δημιουργούμε ένα νέο γράφο με $O(V^2)$ κόμβους και $O(VE)$ ακμές όπου ο κάθε κόμβος είναι της μορφής (θέση Κώστα, θέση Ανδρέα, επόμενη κίνηση) και κάθε ακμή της μορφής ((K,A,K), (K',A,A)) ή ((K,A,A), (K,A',K)), όπου οι (K,K') και (A,A') είναι ακμές του αρχικού γράφου
- Θα δώσουμε σε κάθε κόμβο μια ετικέτα A, K ή I αν ξεκινώντας απ' αυτή την κατάσταση το παιχνίδι καταλήγει σε νίκη του Ανδρέα, του Κώστα ή έρχεται ισόπαλο
- Αν για ένα κόμβο γνωρίζουμε την ετικέτα όλων των γειτόνων του μπορούμε εύκολα να βρούμε τη δική του

Λύση (Ιδέα):

- Έστω ένας κόμβος v στον οποίο παίζει ο Κώστας
- Αν υπάρχει έστω και **ένας** γείτονας του v που έχει ετικέτα K, τότε θα έχουμε label(v) = K
- Αν **όλοι** οι γείτονες του v έχουν ετικέτα A, τότε θα έχουμε label(v) = A
- Διαφορετικά, αν υπάρχουν γείτονες που έχουν ετικέτα A και γείτονες που έχουν ετικέτα I θα έχουμε label(v) = I
- Θα ακολουθήσουμε μια bottom-up προσέγγιση για να δώσουμε τις ετικέτες με τη σωστή σειρά

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση (Ιδέα):

- Έστω ένας κόμβος v στον οποίο παίζει ο Κώστας
- Αν υπάρχει έστω και **ένας** γείτονας του v που έχει ετικέτα K, τότε θα έχουμε label(v) = K
- Αν **όλοι** οι γείτονες του v έχουν ετικέτα A, τότε θα έχουμε label(v) = A
- Διαφορετικά, αν υπάρχουν γείτονες που έχουν ετικέτα A και γείτονες που έχουν ετικέτα I θα έχουμε label(v) = I
- Θα ακολουθήσουμε μια bottom-up προσέγγιση για να δώσουμε τις ετικέτες με τη σωστή σειρά

Λύση (Ιδέα):

- Έστω ένας κόμβος v στον οποίο παίζει ο Κώστας
- Αν υπάρχει έστω και **ένας** γείτονας του v που έχει ετικέτα K, τότε θα έχουμε label(v) = K
- Αν **όλοι** οι γείτονες του v έχουν ετικέτα A, τότε θα έχουμε label(v) = A
- Διαφορετικά, αν υπάρχουν γείτονες που έχουν ετικέτα A και γείτονες που έχουν ετικέτα I θα έχουμε label(v) = I
- Θα ακολουθήσουμε μια bottom-up προσέγγιση για να δώσουμε τις ετικέτες με τη σωστή σειρά

Λύση (Ιδέα):

- Έστω ένας κόμβος v στον οποίο παίζει ο Κώστας
- Αν υπάρχει έστω και **ένας** γείτονας του v που έχει ετικέτα K, τότε θα έχουμε label(v) = K
- Αν **όλοι** οι γείτονες του v έχουν ετικέτα A, τότε θα έχουμε label(v) = A
- Διαφορετικά, αν υπάρχουν γείτονες που έχουν ετικέτα A και γείτονες που έχουν ετικέτα I θα έχουμε label(v) = I
- Θα ακολουθήσουμε μια bottom-up προσέγγιση για να δώσουμε τις ετικέτες με τη σωστή σειρά

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση (Ιδέα):

- Έστω ένας κόμβος v στον οποίο παίζει ο Κώστας
- Αν υπάρχει έστω και **ένας** γείτονας του v που έχει ετικέτα Κ, τότε θα έχουμε label(v) = Κ
- Αν **όλοι** οι γείτονες του v έχουν ετικέτα Α, τότε θα έχουμε label(v) = Α
- Διαφορετικά, αν υπάρχουν γείτονες που έχουν ετικέτα Α και γείτονες που έχουν ετικέτα Ι θα έχουμε label(v) = Ι
- Θα ακολουθήσουμε μια bottom-up προσέγγιση για να δώσουμε τις ετικέτες με τη σωστή σειρά

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση

- Θα διατυπώσουμε τους κανονόνες "χρωματισμού" των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα K, τον χρωματίζουμε K
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα A, τον χρωματίζουμε A
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα I και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι I
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (K) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (A)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Λύση

- Θα διατυπώσουμε τους κανονόνες “χρωματισμού” των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα Κ, τον χρωματίζουμε Κ
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα Α, τον χρωματίζουμε Α
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα Ι και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι Ι
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (Κ) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (Α)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Λύση

- Θα διατυπώσουμε τους κανονόνες "χρωματισμού" των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα Κ, τον χρωματίζουμε Κ
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα Α, τον χρωματίζουμε Α
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα Ι και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι Ι
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (Κ) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (Α)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Λύση

- Θα διατυπώσουμε τους κανονόνες "χρωματισμού" των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα Κ, τον χρωματίζουμε Κ
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα Α, τον χρωματίζουμε Α
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα Ι και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι Ι
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (Κ) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (Α)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Λύση

- Θα διατυπώσουμε τους κανονόνες "χρωματισμού" των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα Κ, τον χρωματίζουμε Κ
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα Α, τον χρωματίζουμε Α
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα Ι και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι Ι
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (Κ) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (Α)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση

- Θα διατυπώσουμε τους κανονόνες "χρωματισμού" των κόμβων που παίζει ο Κώστας (συμμετρικά για τον Ανδρέα)
  - Άμεσος χρωματισμός: Αν ένας κόμβος που παίζει ο Κώστας έχει **τουλάχιστον ένα** γείτονα με ετικέτα Κ, τον χρωματίζουμε Κ
  - Έμμεσος χρωματισμός: Αν σε ένα κόμβο που παίζει ο Κώστας έχουν **όλοι** οι γείτονες ετικέτα Α, τον χρωματίζουμε Α
- Αρχικά όλοι οι κόμβοι έχουν την ετικέτα Ι και για κάθε κόμβο κρατάμε έναν μετρητή των γειτόνων που είναι Ι
- Δίνουμε άμεσα ετικέτα σε όλους τους κόμβους όπου ο Κώστας βρίσκεται στο καταφύγιο (Κ) και ο Ανδρέας βρίσκεται στην ίδια θέση με τον Κώστα (Α)
- Κάθε φορά που δίνουμε ετικέτα σε ένα κόμβο τον τοποθετούμε σε μια ουρά

Λύση

- Μέχρι να αδειάσει η ουρά:
  - Βγάζουμε το πρώτο στοιχείο, ελέγχουμε τους γείτονές του και προσπαθούμε να δώσουμε ετικέτα σε όσους είναι I
  - Ελέγχουμε για κάθε γείτονα αν μπορούμε να κάνουμε άμεσο χρωματισμό
  - Αλλιώς μειώνουμε τον μετρητή του γείτονα που δε μπορούμε να χρωματίσουμε άμεσα κατά 1 και όταν γίνει 0 κάνουμε έμμεσο χρωματισμό
- Ο αλγόριθμος τερματίζει όταν αδειάσει η ουρά, οπότε και δε μπορούμε να κάνουμε άλλους χρωματισμούς

Πολυπλοκότητα Η πολυπλοκότητα του προηγούμενου αλγορίθμου είναι γραμμική στο μέγεθος του νέου γράφου που κατασκευάσαμε, άρα το συνολικό κόστος είναι $O(V^2 + VE)$

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση

- Μέχρι να αδειάσει η ουρά:
  - Βγάζουμε το πρώτο στοιχείο, ελέγχουμε τους γείτονές του και προσπαθούμε να δώσουμε ετικέτα σε όσους είναι I
  - Ελέγχουμε για κάθε γείτονα αν μπορούμε να κάνουμε άμεσο χρωματισμό
  - Αλλιώς μειώνουμε τον μετρητή του γείτονα που δε μπορούμε να χρωματίσουμε άμεσα κατά 1 και όταν γίνει 0 κάνουμε έμμεσο χρωματισμό
- Ο αλγόριθμος τερματίζει όταν αδειάσει η ουρά, οπότε και δε μπορούμε να κάνουμε άλλους χρωματισμούς

Πολυπλοκότητα Η πολυπλοκότητα του προηγούμενου αλγορίθμου είναι γραμμική στο μέγεθος του νέου γράφου που κατασκευάσαμε, άρα το συνολικό κόστος είναι $O(V^2 + VE)$

Άσκηση 3: Κλέφτες και Αστυνόμοι

Λύση

- Μέχρι να αδειάσει η ουρά:
  - Βγάζουμε το πρώτο στοιχείο, ελέγχουμε τους γείτονές του και προσπαθούμε να δώσουμε ετικέτα σε όσους είναι I
  - Ελέγχουμε για κάθε γείτονα αν μπορούμε να κάνουμε άμεσο χρωματισμό
  - Αλλιώς μειώνουμε τον μετρητή του γείτονα που δε μπορούμε να χρωματίσουμε άμεσα κατά 1 και όταν γίνει 0 κάνουμε έμμεσο χρωματισμό
- Ο αλγόριθμος τερματίζει όταν αδειάσει η ουρά, οπότε και δε μπορούμε να κάνουμε άλλους χρωματισμούς

Πολυπλοκότητα Η πολυπλοκότητα του προηγούμενου αλγορίθμου είναι γραμμική στο μέγεθος του νέου γράφου που κατασκευάσαμε, άρα το συνολικό κόστος είναι $O(V^2 + VE)$

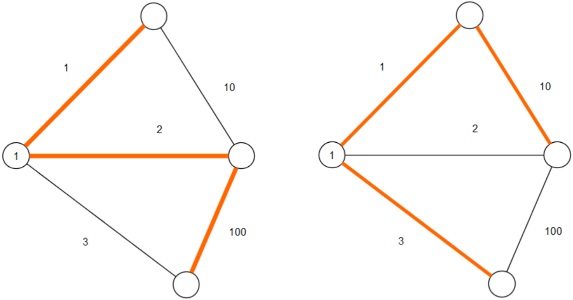
Λύση

- Μέχρι να αδειάσει η ουρά:
  - Βγάζουμε το πρώτο στοιχείο, ελέγχουμε τους γείτονές του και προσπαθούμε να δώσουμε ετικέτα σε όσους είναι I
  - Ελέγχουμε για κάθε γείτονα αν μπορούμε να κάνουμε άμεσο χρωματισμό
  - Αλλιώς μειώνουμε τον μετρητή του γείτονα που δε μπορούμε να χρωματίσουμε άμεσα κατά 1 και όταν γίνει 0 κάνουμε έμμεσο χρωματισμό
- Ο αλγόριθμος τερματίζει όταν αδειάσει η ουρά, οπότε και δε μπορούμε να κάνουμε άλλους χρωματισμούς

Πολυπλοκότητα Η πολυπλοκότητα του προηγούμενου αλγορίθμου είναι γραμμική στο μέγεθος του νέου γράφου που κατασκευάσαμε, άρα το συνολικό κόστος είναι $O(V^2 + VE)$

Ερώτημα Α

Το άπληστο κριτήριο δε λειτουργεί. Για παράδειγμα, σε αυτό το γράφημα με απαιτούμενο βαθμό 2 στον κόμβο 1 θα παίρναμε το αριστερό με βάρος 103, ενώ το ελάχιστο έχει βάρος 14.



Ερώτημα Β

Γενική Ιδέα

- Προσθέτουμε μια σταθερά K σε κάθε ακμή που προσπίπτει στο σχετικό κόμβο.
- Όσο μεγαλώνει το K, τόσο λιγότερες ακμές περιέχει το ΕΣΔ του επαγόμενου γραφήματος του προσπίπτουν στο σχετικό κόμβο.
- Αρκεί να κάνουμε δυαδική αναζήτηση για να βρούμε τη κατάλληλη τιμή του K.

Άσκηση 4 - Ελάχιστο Συνδεκτικό Δέντρο με Περιορισμούς

Πρόταση 1

Για κάθε απαιτούμενο βαθμό το απαιτούμενο ΕΣΔ με περιορισμούς υλοποιείται ως το ΕΣΔ του δέντρου επαυξημένου κατά K.

Πρόταση 2

Τα μοναδικά K στα οποία αλλάζει το ο βαθμός του απαιτούμενου κόμβου στο ΕΣΔ είναι αυτά για τα οποία μια ακμή προσπίπτουσα στον κόμβο γίνεται ίση με μια άλλη ακμή του γράφου όταν αυξηθεί κατά K. Δηλαδή, αρκεί να εξετάσουμε τα $K \in \{e - a : e \in E \wedge a \in A\}$ όπου E το σύνολο των διαφορετικών βαρών ακμών όλου του γραφήματος και A το σύνολο των διαφορετικών βαρών ακμών που προσπίπτουν στον απαιτούμενο κόμβο.

Άσκηση 4 - Ελάχιστο Συνδεκτικό Δέντρο με Περιορισμούς

Απόδειξη 1

Για αρκετά αρνητικό K το ΕΣΔ παίρνει όλες τις σχετικές ακμές, ενώ για αρκετά θετικό παίρνει τις ελάχιστες δυνατές. Συνολικά, όλες οι τιμές ανάμεσα στην ελάχιστη και μέγιστη δυνατή επιτυγχάνονται.

Απόδειξη 2

Αφού για να βγεί από το ΕΣΔ μια ακμή πρέπει να αντικατασταθεί από μια άλλη, σημαίνει πως για κάποιο K γίνεται ίση μια προσπίπτουσα στον απαιτούμενο κόμβο με κάποια άλλη του γραφήματος. Αυτό μπορεί να συμβεί μόνο όταν το K είναι ίσο με τη διαφορά μια προσπίπτουσας ακμής και μιας άλλης ακμής του γραφήματος.

Σημείωση: Σε κάποια βήματα ενδέχεται να βγαίνουν πάνω από μια ακμές από το ΕΣΔ για το ίδιο K.

Άσκηση 4 - Ελάχιστο Συνδεκτικό Δέντρο με Περιορισμούς

Αλγόριθμος

Άρχικά, ταξινομούμε το σύνολο των Κ που εξετάζουμε, δηλααδή $K \in \{e - a : e \in E \wedge a \in A\}$, το οποίο παίρνει χρόνο το πολύ $O(|E| * |A| * log(|E| * |A|))$.
Έπειτα, κάνουμε δυαδική αναζήτηση πάνω σε αυτό το σύνολο. Αυτό γίνεται το πολύ $O(log(|E| * |A|)) \leq O(log(|E| * |E|)) = O(2 * log(|E|))$ φορές. Σε κάθε επανάληψη, προσθέτουμε το συγκεκριμένο Κ στις ακμές που προσπίπτουν στον απαιτούμενο κόμβο και εφαρμόζουμε Kruskal, που παίρνει $O(E * log(|E|))$ κάθε φορά. Αν ο απαιτούμενος βαθμός δεν είναι αρκετός τότε μειώνουμε το Κ και αντίστροφα.
Συνολικά, απαιτείται $O(|E| * |A| * log(|E|) + |E| * log^2(|E|))$

Σημείωση: Φυσικά, και χωρίς να περιορίζαμε το σύνολο των Κ και απλά κάναμε δυαδική αναζήτηση στο $[-_{max}, E_{max}]$, πάλι θα είχαμε πολυωνυμικό φράγμα στον χρόνο.

Άσκηση 5 - Αλγόριθμος Boruvka

Είσοδος: Μη κατευθυνόμενο γράφημα $G(V, E, w)$, με n κορυφές, m ακμές και βάρος $w(e)$ σε κάθε ακμή $e \in \mathbb{E}$, με όλα τα βάρη διαφορετικά μεταξύ τους.

Ερώτημα Α

Ζητούμενα: Για τον αλγόριθμο Boruvka νδο

1. Ο αλγόριθμος καταλήγει πάντα σε συνδετικό δέντρο.
2. Το συνδετικό δέντρο είναι ελαχίστου βάρους.

Ιδέα

Στον αλγόριθμο σε κάθε βήμα ενώνουμε τα συνδεδεμένα κομμάτια ενός δάσους. Για κάθε κομμάτι βρίσκουμε την προσπίπτουσα ακμή ελάχιστου κόστους. Προσθέτουμε τις ακμές στο δάσος και επαναλαμβάνουμε τη διαδικασία.

Άσκηση 5 - Αλγόριθμος Boruvka

Είσοδος: Μη κατευθυνόμενο γράφημα $G(V, E, w)$, με n κορυφές, m ακμές και βάρος $w(e)$ σε κάθε ακμή $e \in \mathbb{E}$, με όλα τα βάρη διαφορετικά μεταξύ τους.

Ερώτημα Α

Ζητούμενα: Για τον αλγόριθμο Boruvka νδο

1. Ο αλγόριθμος καταλήγει πάντα σε συνδετικό δέντρο.
2. Το συνδετικό δέντρο είναι ελαχίστου βάρους.

Ιδέα

Στον αλγόριθμο σε κάθε βήμα ενώνουμε τα συνδεδεμένα κομμάτια ενός δάσους. Για κάθε κομμάτι βρίσκουμε την προσπίπτουσα ακμή ελάχιστου κόστους. Προσθέτουμε τις ακμές στο δάσος και επαναλαμβάνουμε τη διαδικασία.

Άσκηση 5 - Αλγόριθμος Boruvka

1. Έστω ότι το γράφημα που προκύπτει περιέχει κύκλο. Έστω ότι στην τρέχουσα επανάληψη έχουμε εντοπίσει τις k προσπίπτουσες ακμές ελαχίστου κόστους $e_1, e_2, \ldots, e_k$ που σχηματίζουν κύκλο και e_1 η ελάχιστη ακμή. Επομένως πρέπει να έχουμε $w(e_1) < w(e_2) < \cdots < w(e_k) < w(e_1)$ προκειμένου να διαλέξει ο αλγόριθμος και την ακμή e_k.

Άτοπο

Σε κάθε επανάληψη, για κάθε ακμή που επιλέγεται ενώνονται συνεκτικές συνιστώσες.

Έτσι καταλήγουμε πάντα σε ένα συνεκτικό και ακυκλικό γράφο, δηλαδή σε ένα συνδετικό δέντρο.

Άσκηση 5 - Αλγόριθμος Boruvka

1. Έστω ότι το γράφημα που προκύπτει περιέχει κύκλο. Έστω ότι στην τρέχουσα επανάληψη έχουμε εντοπίσει τις k προσπίπτουσες ακμές ελαχίστου κόστους $e_1, e_2, \ldots, e_k$ που σχηματίζουν κύκλο και e_1 η ελάχιστη ακμή. Επομένως πρέπει να έχουμε $w(e_1) < w(e_2) < \cdots < w(e_k) < w(e_1)$ προκειμένου να διαλέξει ο αλγόριθμος και την ακμή e_k.

Άτοπο

Σε κάθε επανάληψη, για κάθε ακμή που επιλέγεται ενώνονται συνεκτικές συνιστώσες.

Έτσι καταλήγουμε πάντα σε ένα συνεκτικό και ακυκλικό γράφο, δηλαδή σε ένα συνδετικό δέντρο.

Άσκηση 5 - Αλγόριθμος Boruvka

1. Έστω ότι το γράφημα που προκύπτει περιέχει κύκλο. Έστω ότι στην τρέχουσα επανάληψη έχουμε εντοπίσει τις k προσπίπτουσες ακμές ελαχίστου κόστους $e_1, e_2, \ldots, e_k$ που σχηματίζουν κύκλο και e_1 η ελάχιστη ακμή. Επομένως πρέπει να έχουμε $w(e_1) < w(e_2) < \cdots < w(e_k) < w(e_1)$ προκειμένου να διαλέξει ο αλγόριθμος και την ακμή e_k.

Άτοπο

Σε κάθε επανάληψη, για κάθε ακμή που επιλέγεται ενώνονται συνεκτικές συνιστώσες.

Έτσι καταλήγουμε πάντα σε ένα συνεκτικό και ακυκλικό γράφο, δηλαδή σε ένα συνδετικό δέντρο.

Άσκηση 5 - Αλγόριθμος Boruvka

2. Έστω ότι το γράφημα που προκύπτει δεν είναι είναι ελαχίστου βάρους. Αυτό σημαίνει ότι υπάρχει ένα δέντρο T (MST) που δεν χρησιμοποιεί τις ελάχιστες προσπίπτουσες ακμές σε κάθε κόμβο. Έστω λοιπόν ότι σε κάποια τομή $(S, \overline{S})$ το MST δεν περιέχει την ακμή ελαχίστου βάρους $e*$. Αν προσθέσουμε την $e*$ στο T τότε δημιουργούμε ένα κύκλο C. Υποχρεωτικά πρέπει να υπάρχει τουλάχιστον μία ακμή που διασχίζει την τομή και ανήκει στον κύκλο C. Για όλες τις ακμές $e \in C \cap (S, \overline{S})$, $e \neq e*$ ισχύει $w(e) > w(e*)$ καθώς όλα τα βάρη είναι μοναδικά. Άρα αν στο δέντρο αντικαταστήσουμε την e με την $e*$ καταλήγουμε σε ένα συνδετικό δέντρο με αυστηρά μικρότερο βάρος.

Άτοπο

Έτσι καταλήγουμε πάντα σε MST.

Άσκηση 5 - Αλγόριθμος Boruvka

2. Έστω ότι το γράφημα που προκύπτει δεν είναι είναι ελαχίστου βάρους. Αυτό σημαίνει ότι υπάρχει ένα δέντρο T (MST) που δεν χρησιμοποιεί τις ελάχιστες προσπίπτουσες ακμές σε κάθε κόμβο. Έστω λοιπόν ότι σε κάποια τομή $(S, \overline{S})$ το MST δεν περιέχει την ακμή ελαχίστου βάρους $e*$. Αν προσθέσουμε την $e*$ στο T τότε δημιουργούμε ένα κύκλο C. Υποχρεωτικά πρέπει να υπάρχει τουλάχιστον μία ακμή που διασχίζει την τομή και ανήκει στον κύκλο C. Για όλες τις ακμές $e \in C \cap (S, \overline{S})$, $e \neq e*$ ισχύει $w(e) > w(e*)$ καθώς όλα τα βάρη είναι μοναδικά. Άρα αν στο δέντρο αντικαταστήσουμε την e με την $e*$ καταλήγουμε σε ένα συνδετικό δέντρο με αυστηρά μικρότερο βάρος.

Άτοπο

Έτσι καταλήγουμε πάντα σε MST.

Άσκηση 5 - Αλγόριθμος Boruvka

2. Έστω ότι το γράφημα που προκύπτει δεν είναι είναι ελαχίστου βάρους. Αυτό σημαίνει ότι υπάρχει ένα δέντρο T (MST) που δεν χρησιμοποιεί τις ελάχιστες προσπίπτουσες ακμές σε κάθε κόμβο. Έστω λοιπόν ότι σε κάποια τομή $(S, \overline{S})$ το MST δεν περιέχει την ακμή ελαχίστου βάρους $e*$. Αν προσθέσουμε την $e*$ στο T τότε δημιουργούμε ένα κύκλο C. Υποχρεωτικά πρέπει να υπάρχει τουλάχιστον μία ακμή που διασχίζει την τομή και ανήκει στον κύκλο C. Για όλες τις ακμές $e \in C \cap (S, \overline{S})$, $e \neq e*$ ισχύει $w(e) > w(e*)$ καθώς όλα τα βάρη είναι μοναδικά. Άρα αν στο δέντρο αντικαταστήσουμε την e με την $e*$ καταλήγουμε σε ένα συνδετικό δέντρο με αυστηρά μικρότερο βάρος.

Άτοπο

Έτσι καταλήγουμε πάντα σε MST.

Άσκηση 5 - Αλγόριθμος Boruvka

2. Έστω ότι το γράφημα που προκύπτει δεν είναι είναι ελαχίστου βάρους. Αυτό σημαίνει ότι υπάρχει ένα δέντρο T (MST) που δεν χρησιμοποιεί τις ελάχιστες προσπίπτουσες ακμές σε κάθε κόμβο. Έστω λοιπόν ότι σε κάποια τομή $(S, \overline{S})$ το MST δεν περιέχει την ακμή ελαχίστου βάρους $e*$. Αν προσθέσουμε την $e*$ στο T τότε δημιουργούμε ένα κύκλο C. Υποχρεωτικά πρέπει να υπάρχει τουλάχιστον μία ακμή που διασχίζει την τομή και ανήκει στον κύκλο C. Για όλες τις ακμές $e \in C \cap (S, \overline{S})$, $e \neq e*$ ισχύει $w(e) > w(e*)$ καθώς όλα τα βάρη είναι μοναδικά. Άρα αν στο δέντρο αντικαταστήσουμε την e με την $e*$ καταλήγουμε σε ένα συνδετικό δέντρο με αυστηρά μικρότερο βάρος.

Άτοπο

Έτσι καταλήγουμε πάντα σε MST.

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Β

Αλγόριθμος

1. Ξεκινάμε με ένα δάσος από απομονωμένους κόμβους
2. Όσο υπάρχουν περισσότερα του ενός δέντρων στο δάσος, διαλέγουμε για κάθε ένα από αυτά την ακτίνα με το μικρότερο βάρος το ενώνει με άλλο δέντρο και την προσθέτουμε στο MST.
3. Η έξοδος είναι το MST.

Το βήμα 2 γίνεται σε $O(m)$, χρησιμοποιώντας δομή union-find. Η δομή θα περιέχει όλους τους κόμβους (αρχικά disjointed) και κάθε σύνολο της είναι ένα δέντρο που έχουμε φτιάξει μέχρι αυτήν την επανάληψη. Έτσι, κάνουμε για κάθε ακμή ένα find στη δομή για να βρούμε για κάθε δέντρο, την ελάχιστη που το συνδέει με άλλο δέντρο. Έπειτα, για κάθε ελάχιστη ακμή που έχουμε βρει, εκτελούμε ένα union, για να συνδέσουμε τα δυο δέντρα που συνδέει.

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Β

Τέλος, σε κάθε εκτέλεση του βήματος 3 το πλήθος των δέντρων υποδιπλασιάζεται (ενώ έχουμε ξεκινήσει αρικά με n δέντρα). Έτσι, ο αλγόριθμος θα κάνει το πολύ $logn$ επαναλήψεις για να καταλήξει στο συνολικό spanning tree. Επομένως, συνολικά απαιτείται χρόνος $O(mlogn)$.

Ερώτημα Γ

Ιδέα

Να τρέξουμε το βήμα 3 του αλγορίθμου για να μειώσουμε τον αριθμό των κόμβων και στη συνέχεια να τρέξουμε τον αλγόριθμο του Prim.

Πόσες φορές όμως πρέπει να τρέξουμε το βήμα 3; Έστω ότι το τρέχουμε r φορές, με κόστος (mr).
Μετά από αυτό ο αριθμός των κόμβων είναι το πολύ $\frac{n}{2^r}$.
Άρα ο αλγόριθμος του Prim θέλει $O(\frac{n}{2^r} \cdot log(\frac{n}{2^r}))$. Συνολικά έχουμε $O(mr + \frac{n}{2^r} \cdot log(\frac{n}{2^r}))$.
Διαλέγουμε το $r = log(logn)$ και καταλήγουμε στο $O(m \cdot loglogn)$

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Γ

Ιδέα

Να τρέξουμε το βήμα 3 του αλγορίθμου για να μειώσουμε τον αριθμό των κόμβων και στη συνέχεια να τρέξουμε τον αλγόριθμο του Prim.

Πόσες φορές όμως πρέπει να τρέξουμε το βήμα 3; Έστω ότι το τρέχουμε r φορές, με κόστος (mr).
Μετά από αυτό ο αριθμός των κόμβων είναι το πολύ $\frac{n}{2^r}$.
Άρα ο αλγόριθμος του Prim θέλει $O(\frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$. Συνολικά έχουμε $O(mr + \frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$.
Διαλέγουμε το $r = log(logn)$ και καταλήγουμε στο $O(\dot{m}loglogn)$

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Γ

Ιδέα

Να τρέξουμε το βήμα 3 του αλγορίθμου για να μειώσουμε τον αριθμό των κόμβων και στη συνέχεια να τρέξουμε τον αλγόριθμο του Prim.

Πόσες φορές όμως πρέπει να τρέξουμε το βήμα 3; Έστω ότι το τρέχουμε r φορές, με κόστος (mr).
Μετά από αυτό ο αριθμός των κόμβων είναι το πολύ $\frac{n}{2^r}$.
Άρα ο αλγόριθμος του Prim θέλει $O(\frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$. Συνολικά έχουμε $O(mr + \frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$.
Διαλέγουμε το $r = log(logn)$ και καταλήγουμε στο $O(\dot{m}loglogn)$

Ερώτημα Γ

Ιδέα

Να τρέξουμε το βήμα 3 του αλγορίθμου για να μειώσουμε τον αριθμό των κόμβων και στη συνέχεια να τρέξουμε τον αλγόριθμο του Prim.

Πόσες φορές όμως πρέπει να τρέξουμε το βήμα 3; Έστω ότι το τρέχουμε r φορές, με κόστος (mr).
Μετά από αυτό ο αριθμός των κόμβων είναι το πολύ $\frac{n}{2^r}$.
Άρα ο αλγόριθμος του Prim θέλει $O(\frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$. Συνολικά έχουμε $O(mr + \frac{n}{2^r}\dot{l}og(\frac{n}{2^r}))$.
Διαλέγουμε το $r = log(logn)$ και καταλήγουμε στο $O(m\dot{l}oglogn)$

Ερώτημα Γ

Ιδέα

Να τρέξουμε το βήμα 3 του αλγορίθμου για να μειώσουμε τον αριθμό των κόμβων και στη συνέχεια να τρέξουμε τον αλγόριθμο του Prim.

Πόσες φορές όμως πρέπει να τρέξουμε το βήμα 3; Έστω ότι το τρέχουμε r φορές, με κόστος (mr).
Μετά από αυτό ο αριθμός των κόμβων είναι το πολύ $\frac{n}{2^r}$.
Άρα ο αλγόριθμος του Prim θέλει $O(\frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$. Συνολικά έχουμε $O(mr + \frac{n}{2^r} \dot{l}og(\frac{n}{2^r}))$.
Διαλέγουμε το $r = log(logn)$ και καταλήγουμε στο $O(m\dot{l}oglogn)$

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Δ
Μπορούμε να εφαρμόσουμε τον αλγόριθμο Boruvka, με την εξής παραλλαγή:

Παραλλαγή Boruvka

Σε κάθε επανάληψη του αλγορίθμου κάνουμε contract τις ακμές που έχουμε επιλέξει (ως ελάχιστες που συνδέουν δυο disjoint δέντρα).

- Φροντίζουμε στην "μετατροπή του γράφου σε κανονικό" μετά από κάθε contraction, να κρατάμε τις ακμές με το ελάχιστο βάρος (όταν έχουμε πολλαπλές ακμές ανάμεσα σε δυο κόμβους).
- Η ορθότητα του αλγορίθμου διατηρείται με προφανή τρόπο, καθώς τώρα σε κάθε επανάληψη, κάθε κόμβος αντιστοιχεί σε ένα δέντρο. (Θυμόμαστε τις ακμές που έχουμε κάνει contract και τις επιστρέφουμε στο τέλος ως MST).

Ερώτημα Δ Πολυπλοκότητα:

- Είναι εύκολο να υλοποιήσουμε το contraction των επιλεγμένων ακμών σε $O(m)$ (χρεάζεται ένα πέρασμα από όλες τις ακμές για να διαλέξουμε τις ελάχιστες σε κάθε multi-edge).
- Αν αρχικά έχουμε E ακμές ξέρουμε ότι στο βήμα i θα υπάρχουν το πολύ $\frac{E}{2^{2i}}$ ακμές.
- Λόγω της κλειστότητας όμως, γνωρίζουμε ότι για κάθε γράφο που προκύπτει σε κάθε βήμα i, με σύνολο κόμβων V_i ισχύει $|E_i| = O(V_i)$.
- Συνεπώς, η πολυπλοκότητα προκύπτει από την αναδρομική σχέση:

$$T(V) = T\left(\frac{V}{2}\right) + O(V) \Rightarrow T(V) = O(V)$$

- Προσέξτε ότι στη γενική περίπτωση αυτή η παραλλαγή χρειάζεται χρόνο $O(V^2)$, καθώς $|E| \leq V^2$.

Ερώτημα Δ Πολυπλοκότητα:

- Είναι εύκολο να υλοποιήσουμε το contraction των επιλεγμένων ακμών σε $O(m)$ (χρειάζεται ένα πέρασμα από όλες τις ακμές για να διαλέξουμε τις ελάχιστες σε κάθε multi-edge).
- Αν αρχικά έχουμε E ακμές ξέρουμε ότι στο βήμα i θα υπάρχουν το πολύ $\frac{E}{2^{2i}}$ ακμές.
- Λόγω της κλειστότητας όμως, γνωρίζουμε ότι για κάθε γράφο που προκύπτει σε κάθε βήμα i, με σύνολο κόμβων V_i ισχύει $|E_i| = O(V_i)$.
- Συνεπώς, η πολυπλοκότητα προκύπτει από την αναδρομική σχέση:

$$T(V) = T\left(\frac{V}{2}\right) + O(V) \Rightarrow T(V) = O(V)$$

- Προσέξτε ότι στη γενική περίπτωση αυτή η παραλλαγή χρειάζεται χρόνο $O(V^2)$, καθώς $|E| \leq V^2$.

Ερώτημα Δ Πολυπλοκότητα:

- Είναι εύκολο να υλοποιήσουμε το contraction των επιλεγμένων ακμών σε $O(m)$ (χρειάζεται ένα πέρασμα από όλες τις ακμές για να διαλέξουμε τις ελάχιστες σε κάθε multi-edge).
- Αν αρχικά έχουμε E ακμές ξέρουμε ότι στο βήμα i θα υπάρχουν το πολύ $\frac{E}{2^{2i}}$ ακμές.
- Λόγω της κλειστότητας όμως, γνωρίζουμε ότι για κάθε γράφο που προκύπτει σε κάθε βήμα i, με σύνολο κόμβων V_i ισχύει $|E_i| = O(V_i)$.
- Συνεπώς, η πολυπλοκότητα προκύπτει από την αναδρομική σχέση:

$$T(V) = T\left(\frac{V}{2}\right) + O(V) \Rightarrow T(V) = O(V)$$

- Προσέξτε ότι στη γενική περίπτωση αυτή η παραλλαγή χρειάζεται χρόνο $O(V^2)$, καθώς $|E| \le V^2$.

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Δ Πολυπλοκότητα:

- Είναι εύκολο να υλοποιήσουμε το contraction των επιλεγμένων ακμών σε $O(m)$ (χρειάζεται ένα πέρασμα από όλες τις ακμές για να διαλέξουμε τις ελάχιστες σε κάθε multi-edge).
- Αν αρχικά έχουμε E ακμές ξέρουμε ότι στο βήμα i θα υπάρχουν το πολύ $\frac{E}{2^{2i}}$ ακμές.
- Λόγω της κλειστότητας όμως, γνωρίζουμε ότι για κάθε γράφο που προκύπτει σε κάθε βήμα i, με σύνολο κόμβων V_i ισχύει $|E_i| = O(V_i)$.
- Συνεπώς, η πολυπλοκότητα προκύπτει από την αναδρομική σχέση:

$$T(V) = T\left(\frac{V}{2}\right) + O(V) \Rightarrow T(V) = O(V)$$

- Προσέξτε ότι στη γενική περίπτωση αυτή η παραλλαγή χρειάζεται χρόνο $O(V^2)$, καθώς $|E| \leq V^2$.

Άσκηση 5 - Αλγόριθμος Boruvka

Ερώτημα Δ Πολυπλοκότητα:

- Είναι εύκολο να υλοποιήσουμε το contraction των επιλεγμένων ακμών σε $O(m)$ (χρειάζεται ένα πέρασμα από όλες τις ακμές για να διαλέξουμε τις ελάχιστες σε κάθε multi-edge).
- Αν αρχικά έχουμε E ακμές ξέρουμε ότι στο βήμα i θα υπάρχουν το πολύ $\frac{E}{2^{2i}}$ ακμές.
- Λόγω της κλειστότητας όμως, γνωρίζουμε ότι για κάθε γράφο που προκύπτει σε κάθε βήμα i, με σύνολο κόμβων V_i ισχύει $|E_i| = O(V_i)$.
- Συνεπώς, η πολυπλοκότητα προκύπτει από την αναδρομική σχέση:

$$T(V) = T\left(\frac{V}{2}\right) + O(V) \Rightarrow T(V) = O(V)$$

- Προσέξτε ότι στη γενική περίπτωση αυτή η παραλλαγή χρειάζεται χρόνο $O(V^2)$, καθώς $|E| \leq V^2$.

Άσκηση 6 - Μονοπάτια Ελάχιστου Bottleneck Κόστους για όλα τα Ζεύγη Κορυφών

Ερώτημα Α

Έστω ένα ζεύγος u, v για το οποίο το μοναδικό μονοπάτι πάνω στο ΕΣΔ δεν αποτελεί ένα μονοπάτι ελαχίστου κόστους. Αυτό σημαίνει ότι υπάρχει ένα άλλο μονοπάτι u, v που δεν ανήκει εξ'ολοκλήρου στο δέντρο και έχει μέγιστη ακμή μικρότερη από τη μέγιστη ακμή του μονοπατιού στο ΕΣΔ. Για να καταλήξουμε σε άτοπο χρησιμοποιούμε το cut property των ΕΣΔ.

Αφαίρουμε την μέγιστη ακμή του μονοπατιού u, v από το ΕΣΔ και αποκτούμε δυο συνεκτικές συνιστώσες. Ώστόσο, από την υπόθεση υπάρχει μονοπάτι με όλες τις ακμές μικρότερες της ακμής που μόλις αφαιρέσαμε. Έπομένως, μπορούμε να ξαναενώσουμε τις δυο συνιστώσες με μικρότερο κόστος που είναι άτοπο αφού υποθέσαμε πως έχουμε ΕΣΔ στην αρχή.

Άσκηση 6 - Μονοπάτια Ελάχιστου Bottleneck Κόστους για όλα τα Ζεύγη Κορυφών

Ερώτημα Β

Καθώς εκτελούμε τον Kruskal θα κρατάμε και το πλήθος των κόμβων σε κάθε συνιστώσα που ενώνουμε. Επειδή κάθε φορά που ενώνονται δυο συνιστώσες η ακμή που τις ενώνει είναι πιο βαριά από όλες τις άλλες, αποτελεί το bottleneck για τους κόμβους των δυο συνιστωσών. Επομένως, η ζητούμε τιμή αυξάνεται κάθε φορά κατά $|set_1| * |set_2| * l(e)$. Η μόνη αλλαγή που απαιτείται είναι να αποθηκεύουμε το πλήθος των κόμβων σε κάθε disjoint set structure.

Η συνολική πολυπλοκότητα είναι ίση με του Kruskal $O(|E| * log(|V|))$.