



Πλοήγηση

 3 δευτ.  512 MB

Υπάρχει ένας **συνδεδεμένος μη κατευθυνόμενος απλός γράφος τύπου κάκτος (cactus graph)**¹ με $N \leq 1000$ κόμβους και M ακμές. Οι κόμβοι του έχουν χρώματα (συμβολίζονται με μη αρνητικούς ακέραιους από 0 έως 1499). Αρχικά, όλοι οι κόμβοι έχουν χρώμα 0. Ένα **προκαθορισμένο (deterministic) ρομπότ χωρίς μνήμη**² εξερευνά τον γράφο κινούμενο από κόμβο σε κόμβο. Πρέπει να επισκεφτεί όλους τους κόμβους τουλάχιστον μία φορά και στη συνέχεια να τερματίσει.

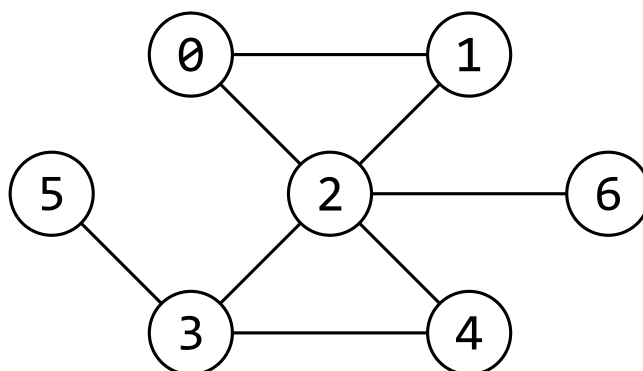
Το ρομπότ ξεκινά από κάποιον κόμβο, ο οποίος θα μπορούσε να είναι οποιοσδήποτε από τους κόμβους του γράφου. Σε κάθε βήμα, βλέπει το χρώμα του τρέχοντος κόμβου και τα χρώματα όλων των γειτονικών κόμβων **σε κάποια σταθερή σειρά για τον τρέχοντα κόμβο** (δηλαδή, εάν επισκεπτεί ξανά τον ίδιο κόμβο θα λάβει την ίδια ακολουθία γειτονικών κόμβων, ακόμα κι αν τα χρώματά τους είναι διαφορετικά από πριν). Το ρομπότ εκτελεί μία από τις ακόλουθες δύο ενέργειες:

1. Αποφασίζει να τερματίσει.
2. Διαλέγει ένα νέο (ή πιθανώς το ίδιο) χρώμα για τον τρέχοντα κόμβο και σε ποιον γειτονικό κόμβο θα μετακινηθεί. Ο γειτονικός κόμβος προσδιορίζεται από έναν δείκτη από 0 έως $D - 1$, όπου D είναι ο αριθμός των γειτονικών κόμβων.

Στη δεύτερη περίπτωση, ο τρέχων κόμβος αλλάζει χρώμα (ή πιθανώς παραμένει το ίδιο χρώμα) και το ρομπότ μετακινείται στον επιλεγμένο γειτονικό κόμβο. Αυτό επαναλαμβάνεται μέχρι το ρομπότ να τερματίσει ή μέχρι να εξαντληθεί το όριο βημάτων. Αν το ρομπότ δεν έχει επισκεφτεί όλους τους κόμβους πριν τερματίσει ή αν ξεπεραστεί το όριο βημάτων χωρίς να τερματίσει, χάνει. Το όριο βημάτων είναι $L = 3000$.

Θα πρέπει να σχεδιάσετε μια στρατηγική για το ρομπότ που να μπορεί να λύσει το πρόβλημα σε οποιονδήποτε τέτοιο γράφο τύπου κάκτος. Επιπλέον, θα πρέπει να προσπαθήσετε να ελαχιστοποιήσετε τον αριθμό των διακριτών χρωμάτων που χρησιμοποιεί η λύση σας. Εδώ το χρώμα 0 πάντα μετράει ως χρησιμοποιημένο.

¹ Ένας **συνδεδεμένος μη κατευθυνόμενος απλός γράφος τύπου κάκτος** είναι ένας συνδεδεμένος μη κατευθυνόμενος απλός γράφος (κάθε κόμβος είναι προσβάσιμος από κάθε άλλο κόμβο· οι ακμές είναι αμφίδρομες· δεν έχει ακμές από έναν κόμβο στον εαυτό του (self-loops) ή επανάληψη ακμών) στον οποίο κάθε ακμή ανήκει σε το πολύ έναν απλό κύκλο (ένας απλός κύκλος είναι ένας κύκλος που περιέχει κάθε κόμβο το πολύ μία φορά). Η ακόλουθη εικόνα παρουσιάζει ένα τέτοιο παράδειγμα.



²Ένα ρομπότ είναι προκαθορισμένο (deterministic) και χωρίς μνήμη, αν η ενέργειά του εξαρτάται μόνο από τις τρέχουσες εισόδους του (δηλαδή δεν αποθηκεύει δεδομένα από βήμα σε βήμα) και πάντα επιλέγει την ίδια ενέργεια όταν του δίνονται οι ίδιες εισοδοί.



Λεπτομέρειες υλοποίησης

Η στρατηγική του ρομπότ θα πρέπει να υλοποιηθεί ως με την ακόλουθη συνάρτηση:

```
std::pair<int, int> navigate(int currColor, std::vector<int> adjColors)
```

Δέχεται ως παραμέτρους το χρώμα του τρέχοντος κόμβου και τα χρώματα όλων των γειτονικών κόμβων (με τη σειρά). Πρέπει να επιστρέφει ένα ζεύγος του οποίου το πρώτο στοιχείο είναι το νέο χρώμα του τρέχοντος κόμβου και το δεύτερο στοιχείο είναι ο δείκτης του γειτονικού κόμβου στον οποίο πρέπει να μετακινηθεί το ρομπότ. Αντίθετα, αν το ρομπότ πρέπει να τερματίσει, η συνάρτηση πρέπει να επιστρέψει το ζεύγος $(-1, -1)$.

Αυτή η συνάρτηση θα κληθεί επαναληπτικά για να επιλέξει τις ενέργειες του ρομπότ. Επειδή είναι προκαθορισμένη, αν η `navigate` κλήθηκε ήδη με κάποιες παραμέτρους, δε θα κληθεί ποτέ ξανά με τις ίδιες παραμέτρους. Αντ'αυτού, θα επαναχρησιμοποιηθεί η προηγούμενη τιμή επιστροφής της. Επιπλέον, κάθε τεστ μπορεί να περιέχει $T \leq 5$ υποτεστ (διαφορετικοί γράφοι και/ή αρχικές θέσεις) και μπορεί να εκτελούνται ταυτόχρονα (δηλαδή το πρόγραμμά σας μπορεί να λαμβάνει εναλλασσόμενες κλήσεις για διαφορετικά υποτεστ). Τέλος, οι κλήσεις στη `navigate` μπορούν να συμβούν σε **ξεχωριστές εκτελέσεις** του προγράμματός σας (αλλά μπορεί επίσης μερικές φορές να συμβούν στην ίδια εκτέλεση). Ο συνολικός αριθμός εκτελέσεων του προγράμματός σας είναι $P = 100$. Λόγω αυτών, το πρόγραμμά σας δεν θα πρέπει να προσπαθεί να περάσει πληροφορίες μεταξύ διαφορετικών κλήσεων.



Περιορισμοί

- $3 \leq N \leq 1000$
- $0 \leq \text{Χρώμα} < 1500$
- $L = 3000$
- $T \leq 5$
- $P = 100$



Βαθμολόγηση

Το κλάσμα S των μονάδων που παίρνετε για ένα υποπρόβλημα εξαρτάται από το C - τον μέγιστο αριθμό διακριτών χρωμάτων που χρησιμοποιεί η λύση σας (συμπεριλαμβανομένου του χρώματος 0) σε οποιοδήποτε τεστ αυτού του υποπροβλήματος ή κάποιου προαπαιτούμενου υποπροβλήματος:

- Αν η λύση σας αποτύχει σε οποιοδήποτε τεστ, τότε $S = 0$.
- Αν $C \leq 4$, τότε $S = 1.0$.
- Αν $4 < C \leq 8$, τότε $S = 1.0 - 0.6 \frac{C-4}{4}$.
- Αν $8 < C \leq 21$, τότε $S = 0.4 \frac{8}{C}$.
- Αν $C > 21$, τότε $S = 0.15$.



Υποπροβλήματα

| Υποπρόβλημα | Μονάδες | Απαιτούμενα υποπροβλ. | N | Επιπλέον Περιορισμοί |
|-------------|---------|-----------------------|------------|---|
| 0 | 0 | — | ≤ 300 | Το παράδειγμα. |
| 1 | 6 | — | ≤ 300 | Ο γράφος είναι ένα κύκλος. ¹ |
| 2 | 7 | — | ≤ 300 | Ο γράφος είναι ένα αστέρι. ² |
| 3 | 9 | — | ≤ 300 | Ο γράφος είναι ένα μονοπάτι. ³ |
| 4 | 16 | 2 – 3 | ≤ 300 | Ο γράφος είναι ένα δένδρο. ⁴ |
| 5 | 27 | — | ≤ 300 | Όλοι οι κόμβοι έχουν το πολύ 3 γειτονικούς κόμβους και ο κόμβος από τον οποίον ξεκινάει το ρομπότ έχει 1 γειτονικό γράφο. |
| 6 | 28 | 0 – 5 | ≤ 300 | — |
| 7 | 7 | 0 – 6 | — | — |

¹Ένας γράφος που είναι κύκλος έχει ακμές: $(i, (i + 1) \bmod N)$ για $0 \leq i < N$.

²Ένας γράφος που είναι αστέρι έχει ακμές: $(0, i)$ για $1 \leq i < N$.

³Ένας γράφος που είναι μονοπάτι έχει ακμές: $(i, i + 1)$ για $0 \leq i < N - 1$.

⁴Ένα δένδρο είναι ένας γράφος χωρίς κύκλους.



Παράδειγμα

Εξετάστε τον απλό γράφο από την εικόνα στην εκφώνηση, ο οποίος έχει $N = 7$, $M = 8$ και ακμές $(0, 1)$, $(1, 2)$, $(2, 0)$, $(2, 3)$, $(3, 4)$, $(4, 2)$, $(3, 5)$ και $(2, 6)$. Επιπλέον, μιας και η σειρά των στοιχείων στις λίστες γειτνίασης των κόμβων έχει σημασία, τις δίνουμε σε αυτόν τον πίνακα:

| Κόμβος | Γειτονικοί κόμβοι |
|--------|-------------------|
| 0 | 2, 1 |
| 1 | 2, 0 |
| 2 | 0, 3, 4, 6, 1 |
| 3 | 4, 5, 2 |
| 4 | 2, 3 |
| 5 | 3 |
| 6 | 2 |

Έστω ότι το ρομπότ ξεκινάει από τον κόμβο 5. Τότε ο επόμενος πίνακας είναι μία πιθανή (αποτυχημένη) ακολουθία από ενέργειες:

| # | Χρώματα | Κόμβος | Κλήση στη <code>navigate</code> | Τιμή επιστροφής |
|---|---------------------|--------|--|-----------------|
| 1 | 0, 0, 0, 0, 0, 0, 0 | 5 | <code>navigate(0, {0})</code> | {1, 0} |
| 2 | 0, 0, 0, 0, 0, 1, 0 | 3 | <code>navigate(0, {0, 1, 0})</code> | {4, 2} |
| 3 | 0, 0, 0, 4, 0, 1, 0 | 2 | <code>navigate(0, {0, 4, 0, 0, 0})</code> | {0, 3} |
| 4 | 0, 0, 0, 4, 0, 1, 0 | 6 | ¹ <code>navigate(0, {0})</code> | {1, 0} |
| 5 | 0, 0, 0, 4, 0, 1, 1 | 2 | <code>navigate(0, {0, 4, 0, 1, 0})</code> | {8, 0} |
| 6 | 0, 0, 8, 4, 0, 1, 1 | 0 | <code>navigate(0, {8, 0})</code> | {3, 0} |
| 7 | 3, 0, 8, 4, 0, 1, 1 | 2 | <code>navigate(8, {3, 4, 0, 1, 0})</code> | {2, 2} |
| 8 | 3, 0, 2, 4, 0, 1, 1 | 4 | <code>navigate(0, {2, 4})</code> | {1, 1} |
| 9 | 3, 0, 2, 4, 1, 1, 1 | 3 | <code>navigate(4, {1, 1, 2})</code> | {-1, -1} |

Εδώ το ρομπότ χρησιμοποίησε ένα σύνολο 6 διακριτών χρωμάτων: 0, 1, 2, 3, 4 και 8 (σημειώστε ότι το 0 θα είχε μετρήσει ως χρησιμοποιημένο ακόμα κι αν το ρομπότ δεν επέστρεφε ποτέ το χρώμα 0, μιας και όλοι οι κόμβοι ξεκινούν με χρώμα 0). Το ρομπότ έτρεξε για 9 επαναλήψεις πριν τερματίσει. Ωστόσο, απέτυχε, μιας και τερμάτισε χωρίς να έχει επισκεφτεί τον κόμβο 1.

¹Σημειώστε πως η κλήση της `navigate` στην επανάληψη 4 δε θα συνέβαινε στην πραγματικότητα. Αυτό συμβαίνει επειδή είναι ισοδύναμη με την κλήση στην επανάληψη 1, οπότε ο βαθμολογητής απλώς θα επαναχρησιμοποιούσε την τιμή επιστροφής της συνάρτησής σας από εκείνη την κλήση. Ωστόσο, αυτό εξακολουθεί να μετράει ως επανάληψη του ρομπότ.

Ενδεικτικός βαθμολογητής (Sample grader)

Ο ενδεικτικός βαθμολογητής δεν τρέχει πολλές φορές το πρόγραμμά σας, οπότε όλες οι κλήσεις στη `navigate` θα είναι στην ίδια εκτέλεση του προγράμματός σας.

Η μορφή εισόδου είναι η εξής: Αρχικά διαβάζεται το T (ο αριθμός των υποτεστ). Στη συνέχεια, για κάθε υποτεστ:

- Γραμμή 1: δύο ακέραιοι - N και M ;
- Γραμμή $2 + i$ (για $0 \leq i < M$): δύο ακέραιοι - A_i και B_i , οι οποίοι είναι οι δύο κόμβοι που ενώνονται με την ακμή i ($0 \leq A_i, B_i < N$).



Ο ενδεικτικός βαθμολογητής θα εκτυπώσει τον αριθμό των διακριτών χρωμάτων που χρησιμοποίησε η λύση σας και τον αριθμό των βημάτων που χρειάστηκε πριν τερματίσει. Εναλλακτικά, θα εκτυπώσει ένα μήνυμα σφάλματος, αν η λύση σας απέτυχε.

Ως προεπιλογή, ο ενδεικτικός βαθμολογητής εκτυπώνει λεπτομερείς πληροφορίες σχετικά με το τι βλέπει και κάνει το ρομπότ σε κάθε επανάληψη. Μπορείτε να απενεργοποιήσετε την προεπιλογή αυτή, αλλάζοντας την τιμή του `DEBUG` από `true` σε `false`.