Σύνδεση υπερδέντρων (supertrees)

Οι κήποι στην προβλήτα (Gardens by the Bay) είναι ένα μεγάλο φυσικό πάρκο στη Σιγκαπούρη. Στο πάρκο αυτό υπάρχουν n πύργοι, γνωστοί ως "υπερδέντρα" supertrees. Οι πύργοι είναι αριθμημένοι από 0 έως και $n-1$. Θέλουμε να κατασκευάσουμε ένα σύνολο από **μηδέν ή περισσότερες** γέφυρες. Κάθε γέφυρα συνδέει ένα ζεύγος διαφορετικών πύργων και μπορεί κανείς να τη διασχίσει **και προς τις δύο** κατευθύνσεις. Δεν πρέπει να υπάρχουν δύο γέφυρες που να συνδέουν το ίδιο ζεύγος πύργων.

Ένα μονοπάτι από τον πύργο x στον πύργο y είναι μία ακολουθία αποτελούμενη από έναν ή περισσότερους πύργους, τέτοια ώστε:

- το πρώτο στοιχείο της ακολουθίας είναι το x,
- το τελευταίο στοιχείο της ακολουθίας είναι το y,
- όλα τα στοιχεία της ακολουθίας είναι **διαφορετικά**, και
- κάθε δύο διαδοχικά στοιχεία (πύργοι) της ακολουθίας συνδέονται μεταξύ τους με μία γέφυρα.

Προσέξτε ότι, εξ ορισμού, υπάρχει ακριβώς ένα μονοπάτι από κάποιον πύργο προς τον εαυτό του και ότι το πλήθος των διαφορετικών μονοπατιών από τον πύργο i προς τον πύργο j είναι το ίδιο με το πλήθος των διαφορετικών μονοπατιών από τον πύργο j προς τον πύργο i.

Ο αρχιτέκτονας του πάρκου θέλει να κατασκευαστούν οι γέφυρες με τέτοιο τρόπο ώστε για κάθε $0 \leq i, j \leq n-1$ να υπάρχουν ακριβώς $p[i][j]$ διαφορετικά μονοπάτια από τον πύργο i προς τον πύργο j, όπου $0 \leq p[i][j] \leq 3$.

Κατασκευάστε ένα σύνολο από γέφυρες που να ικανοποιεί τους περιορισμούς του αρχιτέκτονα, ή αποφασίστε ότι αυτό είναι αδύνατο.

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσετε την παρακάτω συνάρτηση:

```
int construct(int[][] p)
```

- p: ένας πίνακας διαστάσεων $n \times n$ που περιέχει τους περιορισμούς του αρχιτέκτονα.
- Αν η κατασκευή είναι εφικτή, η συνάρτηση αυτή πρέπει να καλέσει ακριβώς μία φορά την `build` (βλ. παρακάτω) για να αναφέρει πώς θα γίνει η κατασκευή, και στη συνέχεια πρέπει να επιστρέψει 1.
- Διαφορετικά, η συνάρτηση πρέπει να επιστρέψει 0 χωρίς να καλέσει καμία φορά την `build`.
- Η συνάρτηση αυτή θα κληθεί ακριβώς μία φορά.

Η συνάρτηση `build` ορίζεται ως εξής:

```
void build(int[][] b)
```

- b: ένας πίνακας διαστάσεων $n \times n$, όπου $b[i][j] = 1$ αν υπάρχει γέφυρα που να συνδέει τους πύργους i και j, ή $b[i][j] = 0$ διαφορετικά.
- Προσέξτε ότι στον πίνακα αυτόν πρέπει να είναι $b[i][j] = b[j][i]$ για κάθε $0 \leq i, j \leq n - 1$ και $b[i][i] = 0$ για κάθε $0 \leq i \leq n - 1$.

Παραδείγματα

Παράδειγμα 1

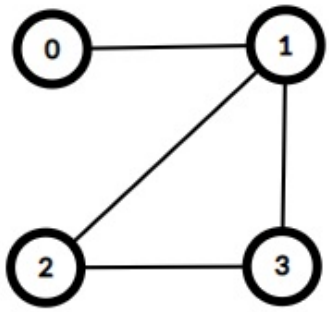
Θεωρήστε την παρακάτω κλήση:

```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Αυτό σημαίνει ότι πρέπει να υπάρχει ακριβώς ένα μονοπάτι από τον πύργο 0 προς τον πύργο 1. Για όλα τα άλλα ζεύγη πύργων (x, y), τέτοια ώστε $0 \leq x < y \leq 3$, πρέπει να υπάρχουν ακριβώς δύο μονοπάτια από τον πύργο x προς τον πύργο y. Αυτό μπορεί να επιτευχθεί με 4 γέφυρες, που να συνδέεουν τα ζεύγη πύργων $(0, 1)$, $(1, 2)$, $(1, 3)$ και $(2, 3)$.

Για να αναφερθεί αυτή τη λύση, η συνάρτηση `construct` πρέπει να καλέσει την `build` ως εξής:

- `build([[0, 1, 0, 0], [1, 0, 1, 1], [0, 1, 0, 1], [0, 1, 1, 0]])`



Στη συνέχεια πρέπει να επιστρέψει 1.

Στην περίπτωση αυτή, υπάρχουν πολλές διαφορετικές κατασκευές που ικανοποιούν τους περιορισμούς του αρχιτέκτονα. Οποιαδήποτε από αυτές τις κατασκευές θεωρείται σωστή.

Παράδειγμα 2

Θεωρήστε την παρακάτω κλήση:

```
construct([[1, 0], [0, 1]])
```

Αυτό σημαίνει ότι δεν πρέπει να υπάρχει τρόπος να μετακινηθεί κανείς μεταξύ των δύο πύργων. Αυτό μπορεί να ικανοποιηθεί αν δεν κατασκευαστεί καμία γέφυρα.

Επομένως, η συνάρτηση `construct` πρέπει να κάνει την εξής κλήση:

- `build([[0, 0], [0, 0]])`

Στη συνέχεια, η `construct` πρέπει να επιστρέψει 1.

Παράδειγμα 3

Θεωρήστε την παρακάτω κλήση:

```
construct([[1, 3], [3, 1]])
```

Αυτό σημαίνει ότι πρέπει να υπάρχουν ακριβώς 3 μονοπάτια από τον πύργο 0 προς τον πύργο 1. Αυτό το σύνολο περιορισμών δεν μπορεί να ικανοποιηθεί. Επομένως, η συνάρτηση `construct` πρέπει να επιστρέψει 0 χωρίς να κάνει καμία κλήση στην `build`.

Περιορισμοί

- $1 \leq n \leq 1000$
- $p[i][i] = 1$ (για κάθε $0 \leq i \leq n-1$)
- $p[i][j] = p[j][i]$ (για κάθε $0 \leq i, j \leq n-1$)
- $0 \leq p[i][j] \leq 3$ (για κάθε $0 \leq i, j \leq n-1$)

Υποπροβλήματα

1. (11 βαθμοί) $p[i][j] = 1$ (για κάθε $0 \leq i, j \leq n-1$)
2. (10 βαθμοί) $p[i][j] = 0$ ή 1 (για κάθε $0 \leq i, j \leq n-1$)
3. (19 βαθμοί) $p[i][j] = 0$ ή 2 (για κάθε $i \neq j, 0 \leq i, j \leq n-1$)
4. (35 βαθμοί) $0 \leq p[i][j] \leq 2$ (για κάθε $0 \leq i, j \leq n-1$) και υπάρχει τουλάχιστον μία κατασκευή που να ικανοποιεί τους περιορισμούς.
5. (21 βαθμοί) $0 \leq p[i][j] \leq 2$ (για κάθε $0 \leq i, j \leq n-1$)
6. (4 βαθμοί) Χωρίς επιπλέον περιορισμούς.

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την είσοδο ως εξής:

- γραμμή 1: n
- γραμμή $2 + i$ ($0 \leq i \leq n-1$): $p[i][0]$ $p[i][1]$ $\ldots$ $p[i][n-1]$

Η έξοδος του υποδειγματικού βαθμολογητή είναι ως εξής:

- γραμμή 1: η τιμή επιστροφής της `construct`.

Αν η τιμή επιστροφής της `construct` είναι 1, ο υποδειγματικός βαθμολογητής τυπώνει επιπλέον:

- γραμμή $2 + i$ ($0 \le i \le n - 1$): $b[i][0]$ $b[i][1]$ $\ldots$ $b[i][n-1]$