توصيل الأشجار الخارقة (supertrees)

حدائق الخليج هي حديقة طبيعية كبيرة في سنغافورة. تحوي الحديقة على n برج، تعرف باسم الأشجار الخارقة. هذه الأبراج مرقمة من 0 إلى $n-1$. نرغب بإنشاء مجموعة من الجسور عددها 0 أو أكثر. كل جسر يربط بين زوج من الأبراج المختلفة ويمكن عبوره في كلا الإتجاهين. لا يمكن إنشاء جسرين يربطان بين نفس الزوج من الأبراج.

الطريق من البرج x إلى البرج y هو سلسلة مكونة من برج واحد أو أكثر بحيث:

- أول عنصر في السلسلة هو x،
- آخر عنصر في السلسلة هو y،
- كل عناصر السلسلة **مختلفة عن بعضها البعض**،
- يوجد جسر يربط بين كل برجين متتاليين في السلسلة.

لاحظ أنه حسب التعريف، يوجد طريق واحد تماماً من البرج إلى نفسه، وعدد الطرق المختلفة من البرج i إلى البرج j يساوي عدد الطرق المختلفة من البرج j إلى البرج i.

يرغب المهندس المسؤول عن التصميم بإنشاء مجموعة من الجسور ليصبح هناك $p[i][j]$ طريق مختلف من البرج i إلى البرج j من أجل كل $0 \le i, j \le n-1$ بحيث $0 \le p[i][j] \le 3$.

يطلب منك إنشاء مجموعة من الجسور التي تحقق متطلبات المهندس المعماري، أو بيان أن ذلك مستحيل.

تفاصيل التنجيز

عليك تنجيز الإجرائية التالية:

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

- p: مصفوفة أبعادها $n \times n$ تمثل متطلبات المهندس المعماري.
- في حال كان الإنشاء ممكناً، يجب على هذه الإجرائية أن تستدعي الإجرائية `build` (إنظر أدناه) مرة واحدة تماماً للإبلاغ عن طريقة الإنشاء، وبعد ذلك يجب أن تعيد القيمة 1.
- ما عدا ذلك، يجب أن تعيد الإجرائية القيمة 0 بدون أي استدعاء للإجرائية `build`.
- سيتم استدعاء هذه الإجرائية مرة واحدة تماماً.

الإجرائية `build` معرفة كما يلي:

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

- b: مصفوفة أبعادها $n \times n$، حيث $b[i][j] = 1$ إذا تم انشاء جسر يربط بين البرجين i و j، وإلا تكون $b[i][j] = 0$.

- لاحظ أن المصفوفة b يجب أن تحقق أن $b[i][j] = b[j][i]$ من أجل كل $0 \leq i, j \leq n-1$ و $b[i][i] = 0$ من أجل كل $0 \leq i \leq n-1$.

الأمثلة

المثال 1

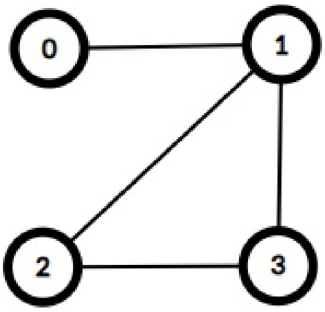
ليكن الاستدعاء التالي:

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

هذا يعني أنه يجب أن يكون هنالك طريق واحد تماماً من البرج 0 إلى البرج 1. يجب أن يكون هنالك حصراً طريقين مختلفين من البرج x إلى البرج y من أجل كل أزواج الأبراج المتبقية (x, y) حيث $0 \leq x < y \leq 3$. يمكن تحقيق ذلك باستخدام 4 جسور تصل بين أزواج الأبراج $(0, 1)$، $(1, 2)$، $(1, 3)$ و $(2, 3)$.

للإبلاغ عن هذا الحل، يجب أن تقوم الإجرائية `construct` بالاستدعاء التالي:

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



وبعدها يجب أن تعيد القيمة 1.

في هذه الحالة، يوجد أكثر من طريقة لإنشاء الجسور بحيث تحقق هذه المتطلبات، سيتم اعتبارها كلها صحيحة.

المثال 2

ليكن الاستدعاء التالي:

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

هذا يعني أنه من الواجب عدم وجود أي طريق يصل بين البرجين. ويمكن تحقيق ذلك فقط من خلال عدم إنشاء أي جسر. وبالتالي، يجب أن تقوم الإجرائية `construct` بالاستدعاء التالي:

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

بعد ذلك يجب أن تعيد الإجرائية `construct` القيمة 1.

المثال 3

ليكن الاستدعاء التالي:

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

هذا يعني أنه من الواجب أن يكون هنالك 3 طرق تماماً من البرج 0 إلى البرج 1. هذه الشروط لا يمكن تحقيقها. لذلك، يجب أن تعيد الإجرائية `construct` القيمة 0 دون أي استدعاء للإجرائية `build`.

القيود

- $1 \le n \le 1000$
- $p[i][i] = 1$ (من أجل كل $0 \le i \le n-1$)
- $p[i][j] = p[j][i]$ (من أجل كل $0 \le i, j \le n-1$)
- $0 \le p[i][j] \le 3$ (من أجل كل $0 \le i, j \le n-1$)

المسائل الجزئية

1. (11 علامة) $p[i][j] = 1$ (لكل $0 \le i, j \le n-1$)
2. (10 علامات) $p[i][j]$ تساوي 0 أو 1 (لكل $i \neq j$، $0 \le i, j \le n-1$)
3. (19 علامة) $p[i][j]$ تساوي 0 أو 2 (لكل $i \neq j$، $0 \le i, j \le n-1$)
4. (35 علامة) $0 \le p[i][j] \le 2$ (لكل $0 \le i, j \le n-1$) ويوجد على الأقل طريقة واحدة لإنشاء جسور تحقق كل المتطلبات.
5. (21 علامة) $0 \le p[i][j] \le 2$ (لكل $0 \le i, j \le n-1$)
6. (4 علامات) لا يوجد قيود إضافية.

المصحح النموذجي

المصحح النموذجي يقرأ الدخل بالصيغة التالية:

- السطر 1: n
- الأسطر $2+i$ ($0 \le i \le n-1$): $p[i][0]\ p[i][1]\ \ldots\ p[i][n-1]$

خرج المصحح النموذجي هو بالصيغة التالية:

- السطر 1: القيمة التي تعيدها الإجرائية `construct`.

اذا كانت القيمة التي تعيدها الإجرائية `construct` تساوي 1، يقوم المصحح النموذجي بطباعة الخرج الإضافي التالي:

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