

## حدائق النافورة

في حديقة قريبة هناك  $n$  نوافير، مرقمة من 0 إلى  $n - 1$ . سنمثل النوافير كنقاط في مستوى ثنائي الأبعاد. لذا، النافورة  $i$  (  $0 \leq i \leq n - 1$  ) هي نقطة  $(x[i], y[i])$  حيث أن  $x[i]$  و  $y[i]$  هي أرقام صحيحة زوجية. أماكن النوافير كلها مختلفة.

تيموثي المهندس المعماري تم توظيفه لتوضع الخطة لبناء بعض الطرق ووضع مقعد لكل طريق. الطريق هو خط أفقي أو رأسي بطول 2، حيث أن طرفاه هما نافورتان مختلفتان. يجب أن تصمم الطرق بحيث يستطيع الشخص أن يسافر بين أي نافورتين بالتحرك على الطرق. في البداية، لا يوجد أي طرق في الحديقة.

لكل طريق، يجب أن يتم تخصيص (أي أن يواجه) مقعد واحد بالضبط هذا الطريق. كل مقعد يجب أن يوضع في نقطة  $(a, b)$  حيث أن  $a$  و  $b$  هم أرقام صحيحة فردية. أماكن المقاعد كلها يجب أن تكون مختلفة. المقعد الذي عند  $(a, b)$  يمكن أن يخصص لطرق فقط إذا كان كلا نهايتي الطريق إحدى النقاط  $(a - 1, b + 1)$ ،  $(a - 1, b - 1)$ ،  $(a + 1, b + 1)$  و  $(a + 1, b - 1)$ . كمثال، المقعد الذي عند  $(3, 3)$  يمكن أن يخصص لطريق من هذه الأربع قطع مستقيمة  $(2, 2) - (2, 4)$ ،  $(2, 4) - (4, 4)$ ،  $(4, 4) - (4, 2)$ ،  $(4, 2) - (2, 2)$ .

ساعد تيموثي في تحديد إذا كان هناك طريقة لبناء الطرق ووضع المقاعد وتخصيصها بحيث تحقق جميع الشروط المذكورة بالأعلى. إذا كان هناك طريقة فعليك تقديم أحد الحلول. إذا كان هناك أكثر من حل يحقق كل الشروط يمكنك تقديم أي منها.

## تفاصيل التجيز

عليك تجيز الإجرائية التالية:

```
int construct_roads(int[] x, int[] y)
```

- $x, y$ : مصفوفتان بحجم  $n$ . لكل  $i$  ( $0 \leq i \leq n - 1$ )، النافورة  $i$  هي نقطة  $(x[i], y[i])$ ، حيث أن  $x[i]$  و  $y[i]$  أرقام صحيحة زوجية.
- إذا كان البناء ممكناً فعلى هذه الإجرائية نداء build (انظر للأسفل) لتقديم الحل ثم عليها إرجاع 1.
- غير ذلك، على الإجرائية إرجاع 0 دون أن تقوم بأي نداء لـ build.
- سيتم نداء هذه الإجرائية مرة واحدة فقط.

يمكن لإجرائيتك مناداة الإجرائية التالية لتقديم بناء للطرق ووضع للمقاعد:

```
void build(int[] u, int[] v, int[] a, int[] b)
```

- افترض أن  $m$  هو عدد الطرق التي سيتم بناؤها.
- $u, v$ : مصفوفتان بحجم  $m$ ، تعبران عن الطرق التي سيتم بناؤها. هذه الطرق مرقمة من 0 إلى  $m - 1$ . لكل  $j$  (  $0 \leq j \leq m - 1$  )، الطريق  $j$  يوصل النافورتين  $u[j]$  و  $v[j]$ . كل طريق يجب أن يكون خطاً أفقياً أو رأسياً بطول 2.
- كل طريقين مختلفين يمكن أن يلتقيا على الأكثر في نقطة واحدة (وهي نافورة). بعد أن يتم بناء الطرق يجب أن يكون ممكناً السفر بين أي نافورتين بالتحرك على الطرق.
- $a, b$ : مصفوفتين بالحجم  $m$ ، تمثلان المقاعد. لكل  $j$  ( $0 \leq j \leq m - 1$ )، هناك مقعد موجود في  $(a[j], b[j])$ ، ومخصص للطريق  $j$ . لا يجب أن يكون هناك مقعدين مختلفين موضوعين في نفس المكان.

## أمثلة

### مثال 1

ليكن لدينا الاستدعاء التالي:

```
construct_roads([4, 4, 6, 4, 2], [4, 6, 4, 2, 4])
```

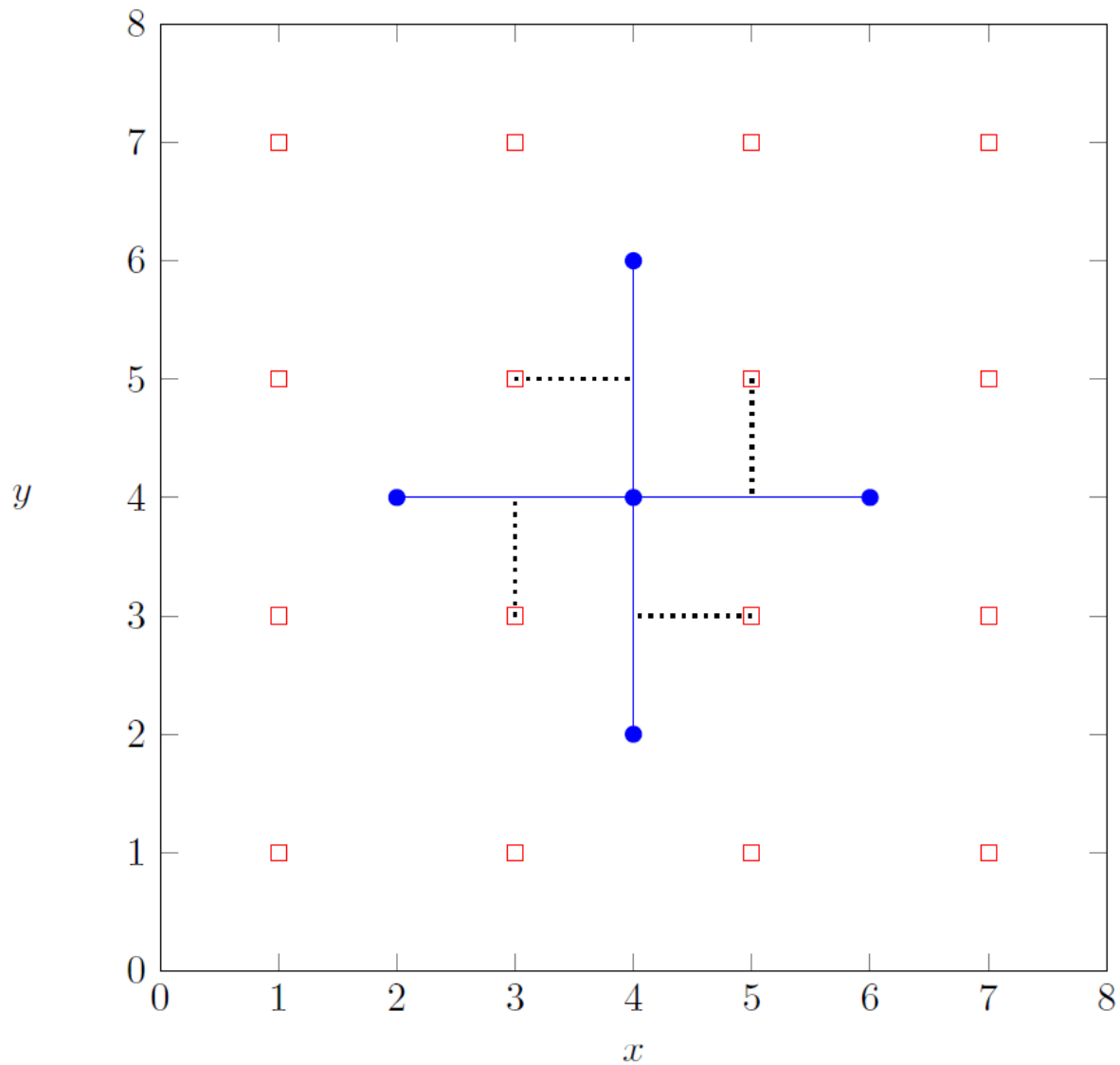
هذا يعني أن هناك 5 نوافير:

- النافورة 0 موجودة في (4, 4) ,
- النافورة 1 موجودة في (4, 6) ,
- النافورة 2 موجودة في (6, 4) ,
- النافورة 3 موجودة في (4, 2) ,
- النافورة 4 موجودة في (2, 4) .

يمكن أن يتم بناء 4 طرف, حيث يوصل كل طريق بين نافورتين, ويمكن وضع المقاعد التالية:

مكان المقعد المخصص	أرقام النوافير التي يوصلها الطريق	رقم الطريق
(5, 5)	0, 2	0
(3, 5)	0, 1	1
(5, 3)	3, 0	2
(3, 3)	4, 0	3

الحل يعبر عنه الشكل التالي:



لتقدين هذا الحل, `construct_roads` يجب أن تقوم بهذا النداء:

```
build([0, 0, 3, 4], [2, 1, 0, 0], [5, 3, 5, 3], [5, 5, 3, 3]) •
```

يجب أن تعيد 1.

لاحظ أنه في هذه الحالة هناك عدة حلول تحقق الشروط, كل هذه الحلول تعتبر صحيحة. كمثال يمكن أيضا عمل هذا النداء `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` ثم إرجاع 1.

مثال 2

ليكن لدينا الاستدعاء التالي:

```
construct_roads([2, 4], [2, 6])
```

النافورة 0 موجودة في (2, 2) والنافورة 1 موجودة في (4, 6). حيث أنه لا طريقة لبناء طرق تحقق الشروط, `construct_roads` يجب أن ترجع 0 دون عمل أي نداء لـ `build`.

## القيود

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$  (لكل  $0 \leq i \leq n - 1$ )
- $x[i]$  و  $y[i]$  أرقام صحيحة زوجية (لكل  $0 \leq i \leq n - 1$ ).
- ليس هناك نافورتين في نفس المكان

## المسائل الجزئية

1. (5 علامات)  $x[i] = 2$  (لكل  $0 \leq i \leq n - 1$ )
2. (10 علامات)  $2 \leq x[i] \leq 4$  (لكل  $0 \leq i \leq n - 1$ )
3. (15 علامة)  $2 \leq x[i] \leq 6$  (لكل  $0 \leq i \leq n - 1$ )
4. (20 points) هناك طريقة واحدة لبناء الطريق بحيث يمكن التنقل بين أي نافورتين باستخدام الطرق.
5. (20 points) لا يوجد أربع نوافير تمثل أربعة أركان لـ  $2 \times 2$  مربع.
6. (30 points) لا يوجد قيود إضافية.

## المصحح النموذجي

يقرأ المصحح النموذجي الدخل وفقًا للتنسيق التالي:

- السطر 1:  $n$
- السطر  $i + 2$  ( $0 \leq i \leq n - 1$ ):  $x[i] \ y[i]$

يطبع المصحح النموذجي بهذا التنسيق:

- السطر 1: القيمة الراجعة من `construct_roads`

إذا كان القيمة الراجعة من `construct_roads` هي 1 و `build(u, v, a, b)` تمت مناداتها، سيطبع المصحح إضافيًا:

- السطر 2:  $m$
- السطر  $j + 3$  ( $0 \leq j \leq m - 1$ ):  $u[j] \ v[j] \ a[j] \ b[j]$