

З'єднання Супердерев (supertrees)

Сади біля затоки - великий парк у Сінгапурі. У парку n веж, які називають супердеревами. Вежі пронумеровано від 0 до $n - 1$. Ми хочемо сконструювати множину з **не менше ніж нуль** мостів. Кожен міст з'єднує пару різних веж, по ньому можна рухатись в **обох** напрямках. Жодні два мости не з'єднують одну і ту-ж пару веж.

Шлях з башти x до башти y - це послідовність з одної, або кількох веж така, що:

- перший елемент послідовності x ,
- останній елемент послідовності y ,
- всі елементи послідовності **різні**, і
- кожен два послідовні елементи послідовності (башти) з'єднані мостом.

Зауважте, що за означенням існує єдиний шлях від башти до самої себе, і кількість різних шляхів від башти i до башти j така сама, як кількість шляхів від башти j до башти i .

Головний архітектор хоче, щоб мости були побудовані таким чином, щоб для всіх $0 \leq i, j \leq n - 1$ було рівно $p[i][j]$ різних шляхів від башти i до башти j , де $0 \leq p[i][j] \leq 3$.

Побудуйте набір мостів, який задовольняє вимоги архітектора, або встановіть, що це неможливо.

Деталі реалізації

Вам потрібно реалізувати наступну процедуру:

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

- p : масив $n \times n$ визначає вимоги архітектора.
- Якщо побудова можлива, процедура повинна здійснити рівно один виклик `build` (дивись нижче) для запуску будівництва, після чого повернути 1.
- В іншому випадку, процедура повинна повернути 0 без жодного виклику `build`.
- Ця процедура викликається рівно один раз.

Визначимо процедуру `build` наступним чином:

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

- b : масив $n \times n$, де $b[i][j] = 1$ якщо вежі i та j з'єднані мостом. В іншому випадку

$$b[i][j] = 0.$$

- Зауважте, що масив має задовольняти умову $b[i][j] = b[j][i]$ для всіх $0 \leq i, j \leq n - 1$ і $b[i][i] = 0$ для всіх $0 \leq i \leq n - 1$.

Приклади

Приклад 1

Розглянемо наступний виклик:

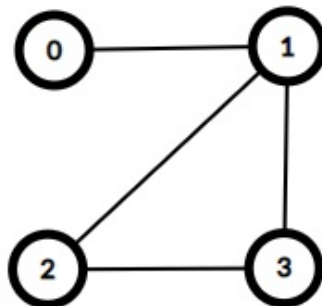
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

Це означає, що має існувати рівно один шлях з башти 0 до башти 1. Для всіх інших пар башт (x, y) , таких, що $0 \leq x < y \leq 3$, має бути рівно два шляхи з башти x до башти y .

Цього можна досягнути побудовою 4 мостів, що з'єднують пари веж $(0, 1)$, $(1, 2)$, $(1, 3)$ та $(2, 3)$.

Щоб повідомити про знаходження розв'язку, процедура `construct` повинна зробити наступний виклик:

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



А потім повернути 1.

В цьому випадку існує кілька конструкцій, що задовольняють задані вимоги, всі вони зараховуються як правильні.

Приклад 2

Розглянемо наступний виклик:

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

Це означає, що не повинно бути маршруту між двома вежами. Цього можна досягти лише якщо не будувати жодних мостів.

Таким чином процедура `construct` повинна зробити наступний виклик:

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

Після чого повернути 1.

Приклад 3

Розглянемо наступний виклик:

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

Це означає, що має бути рівно 3 шляхи з башти 0 до башти 1. Цей набір вимог неможливо задовольнити. Таким чином, процедура `construct` повинна повернути 0 без жодних викликів `build`.

Обмеження

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

Підзадачі

1. (11 балів) $p[i][j] = 1$ (для всіх $0 \leq i, j \leq n - 1$)
2. (10 балів) $p[i][j] = 0$ або 1 (для всіх $0 \leq i, j \leq n - 1$)
3. (19 балів) $p[i][j] = 0$ або 2 (для всіх $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 балів) $0 \leq p[i][j] \leq 2$ (для всіх $0 \leq i, j \leq n - 1$) і існує принаймні одна конструкція, що задовольняє обмеження.
5. (21 бал) $0 \leq p[i][j] \leq 2$ (for all $0 \leq i, j \leq n - 1$)
6. (4 бали) жодних додаткових обмежень.

Приклад модуля перевірки

Модуль перевірки зчитує вхідні дані у наступному форматі:

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

Вивід модуля перевірки має наступний формат:

- рядок 1: значення, що повертає `construct`.

Якщо `construct` повертає значення 1, модуль перевірки додатково виводить:

- рядок $2 + i$ ($0 \leq i \leq n - 1$): $b[i][0] \ b[i][1] \ \dots \ b[i][n - 1]$