

Соединение сверхдеревьев (supertrees)

Сады у Залива — большой природный парк в Сингапуре. В парке расположены n башен, известных как сверхдеревья. Эти башни пронумерованы от 0 до $n - 1$. Мы хотим построить множество из **0 или более** мостов. Каждый мост соединяет пару различных башен и может быть пройден в **любом** направлении. Никакие два моста не должны соединять одну и ту же пару башен.

Путем от башни x до башни y называется последовательность из одной или нескольких башен, для которой выполнены следующие условия:

- первый элемент последовательности — x ,
- последний элемент последовательности — y ,
- последовательность состоит из **различных** элементов,
- любые две последовательные башни в последовательности соединены мостом.

Обратите внимание, что по определению существует единственный путь от башни до нее самой, а число различных путей от башни i до башни j совпадает с числом различных путей от башни j до башни i .

Главный архитектор, ответственный за дизайн, решил, что мосты должны быть построены таким образом, что для всех $0 \leq i, j \leq n - 1$ существует в точности $p[i][j]$ различных путей от башни i до башни j , где $0 \leq p[i][j] \leq 3$.

Постройте множество мостов, удовлетворяющее требованиям архитектора, или определите, что это невозможно.

Детали реализации

Вы должны реализовать следующую функцию:

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

- p : массив размера $n \times n$, содержащий требования архитектора.
- Если построить мосты требуемым образом возможно, эта функция должна совершить ровно один вызов `build` (смотрите ниже) для того, чтобы сообщить, как именно мосты должны быть построены, после чего функция должна вернуть 1.
- В противном случае, функция должна вернуть 0 без каких-либо вызовов функции `build`.
- Функция будет вызвана ровно один раз.

Функция `build` объявлена следующим образом:

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

- b : массив размера $n \times n$, где $b[i][j] = 1$ если существует мост, соединяющий башни i и j , и $b[i][j] = 0$ в противном случае.
- Обратите внимание, что массив b должен удовлетворять условиям $b[i][j] = b[j][i]$ для всех $0 \leq i, j \leq n - 1$ и $b[i][i] = 0$ для всех $0 \leq i \leq n - 1$.

Примеры

Пример 1

Рассмотрим следующий вызов функции:

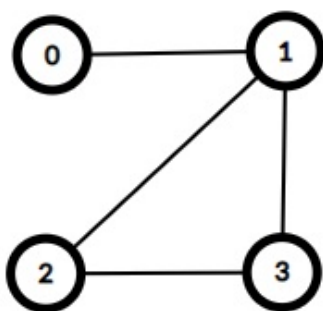
```
construct([[1, 1, 2, 2], [1, 1, 2, 2], [2, 2, 1, 2], [2, 2, 2, 1]])
```

В этом примере требуется, чтобы существовал ровно 1 путь от башни 0 до башни 1. Для каждой из остальных пар башен (x, y) таких, что $0 \leq x < y \leq 3$, должно существовать ровно два пути от башни x до башни y .

Требуемой конфигурации можно достичь с помощью 4 мостов, соединяющих пары башен $(0, 1)$, $(1, 2)$, $(1, 3)$ и $(2, 3)$.

Для того, чтобы сообщить описанную выше конфигурацию мостов, функция `construct` должна совершить следующий вызов функции:

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



После этого функция `construct` должна вернуть 1.

В данном примере существует несколько возможных ответов, удовлетворяющих всем требованиям архитектора, любой из которых будет считаться верным.

Пример 2

Рассмотрим следующий вызов функции:

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

В этом примере требуется, чтобы ни от одной башни до другой не существовало пути. Требуемой конфигурации можно достичь только полным отсутствием мостов.

Для этого функция `construct` должна совершить следующий вызов функции:

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

После этого функция `construct` должна вернуть 1.

Пример 3

Рассмотрим следующий вызов функции:

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

В этом примере требуется, чтобы существовало 3 различных пути от башни 0 до башни 1. Не существует конфигурации мостов, которая удовлетворяла бы этим требованиям. В этом случае функция `construct` должна вернуть 0 без каких-либо вызовов функции `build`.

Ограничения

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

Подзадачи

1. (11 баллов) $p[i][j] = 1$ (для всех $0 \leq i, j \leq n - 1$)
2. (10 баллов) $p[i][j] = 0$ или 1 (для всех $0 \leq i, j \leq n - 1$)
3. (19 баллов) $p[i][j] = 0$ или 2 (для всех $i \neq j, 0 \leq i, j \leq n - 1$)
4. (35 баллов) $0 \leq p[i][j] \leq 2$ (для всех $0 \leq i, j \leq n - 1$) и существует хотя бы одна конфигурация мостов, удовлетворяющая всем требованиям.
5. (21 балл) $0 \leq p[i][j] \leq 2$ (для всех $0 \leq i, j \leq n - 1$)
6. (4 балла) Без дополнительных ограничений.

Пример проверяющего модуля

Проверяющий модуль считывает данные в следующем формате:

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

Проверяющий модуль выводит данные в следующем формате:

- строка 1: возвращаемое значение функции `construct`.

Если возвращаемое значение `construct` равно 1, проверяющий модуль выводит также:

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