

חיבור עצי-על (supertrees)

גנים לצד המפרץ (Gardens by the Bay) הוא פארק טבע גדול בסינגפור. בפארק ישנם n מגדלים, הידועים כעצי-על. המגדלים הללו ממוספרים מ-0 עד $n - 1$. היינו רוצים לבנות קבוצה של **אפס או יותר** גשרים. כל גשר מחבר זוג מגדלים שונים וניתן למעבר **בכל כיוון**. אסור ששני גשרים שונים יחברו את אותו זוג המגדלים.

מסלול ממגדל x למגדל y הוא רצף של מגדל אחד או יותר כך ש:

- האיבר הראשון ברצף הוא x ,
- האיבר האחרון ברצף הוא y ,
- כל האיברים ברצף הם **יחודיים**, ו
- כל שני איברים (מגדלים) עוקבים מחוברים על ידי גשר.

שימו לב שלפי ההגדרה קיים מסלול אחד בדיוק ממגדל לעצמו וכן שמספר המסלולים ממגדל i למגדל j זהה למספר המסלולים ממגדל j למגדל i .

האדריכל הראשי האחראי על העיצוב מעוניין שהגשרים ייבנו כך שלכל $0 \leq i, j \leq n - 1$ קיימים בדיוק $p[i][j]$ מסלולים שונים ממגדל i למגדל j , כאשר $0 \leq p[i][j] \leq 3$.

עליכם לבנות קבוצה של גשרים שעונה על הדרישות של האדריכל, או לפסוק שזה בלתי אפשרי.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

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

- p : מטריצה בגודל $n \times n$ המייצגת את דרישות האדריכל.
- אם הבנייה אפשרית, הפונקציה צריכה לקרוא בדיוק פעם אחת לפונקציה `build` (ראה מטה) כדי לדווח על הבנייה, ולאחר מכן על הפונקציה להחזיר 1.
- אחרת, הפונקציה צריכה להחזיר 0 מבלי לקרוא כלל ל-`build`.
- פונקציה זו נקראת בדיוק פעם אחת.

הפונקציה `build` מוגדרת כך:

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

- b : מטריצה בגודל $n \times n$, כאשר $b[i][j] = 1$ אם קיים גשר המחבר את מגדל i למגדל j , או $b[i][j] = 0$ אחרת.

- שים לב שהמטריצה חייבת לספק את התנאי $b[i][j] = b[j][i]$ לכל $0 \leq i, j \leq n - 1$ וכן ש-
 $b[i][i] = 0$ לכל $0 \leq i \leq n - 1$.

דוגמאות

דוגמה 1

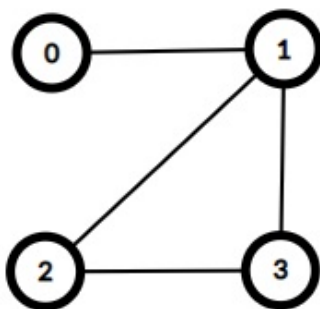
הביטו בקריאה הבאה:

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

צריך להיות קיים בדיוק מסלול אחד ממגדל 0 למגדל 1. לכל שאר זוגות המגדלים (x, y) , כש-
 $0 \leq x < y \leq 3$, צריכים להיות בדיוק שני מסלולים ממגדל x למגדל y . ניתן להשיג זאת באמצעות 4 גשרים,
המחברים את הזוגות $(0, 1)$, $(1, 2)$, $(1, 3)$ ו- $(2, 3)$.

על מנת לדווח על פתרון זה, על הפונקציה ה-construct לבצע את הקריאה הבאה:

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



ואז להחזיר 1.

במקרה זה, ישנן מספר בניות שמתאימות לדרישות, כולן יחשבו נכונות.

דוגמה 2

הביטו בקריאה הבאה:

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

המשמעות כאן היא שלא צריכה להיות אפשרות לעבור בין שני המגדלים. הדרך היחידה לספק את האילוצים
היא על ידי אפס גשרים.

לכן, על הפונקציה construct לבצע את הקריאה הבאה:

• `build([[0, 0], [0, 0]])`

ולאחר מכן, על הפונקציה construct להחזיר 1.

דוגמה 3

הביטו בקריאה הבאה:

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

המשמעות כאן היא שצריכים להיות בדיוק 3 מסלולים ממגדל 0 למגדל 1. לא ניתן לענות על קבוצת הדרישות הזו. לכן, הפונקציה construct צריכה להחזיר 0 ללא אף קריאה ל-build.

מגבלות

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

תתי משימות

1. $p[i][j] = 1$ (11 נקודות) (לכל $0 \leq i, j \leq n - 1$)
2. $1 \text{ or } p[i][j] = 0$ (10 נקודות) (לכל $0 \leq i, j \leq n - 1$)
3. $2 \text{ or } p[i][j] = 0$ (19 נקודות) (לכל $0 \leq i, j \leq n - 1, i \neq j$)
4. $0 \leq p[i][j] \leq 2$ (35 נקודות) (לכל $0 \leq i, j \leq n - 1$) וקיימת לפחות בנייה אחת המספקת את הדרישות.
5. $0 \leq p[i][j] \leq 2$ (21 נקודות) (לכל $0 \leq i, j \leq n - 1$)
6. (4 נקודות) ללא מגבלות נוספות.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא (השורות נקראות משמאל לימין):

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

הפלט של הגריידר לדוגמה הוא בפורמט הבא:

- שורה 1: ערך החזרה של construct.

אם ערך החזרה של construct הוא 1, הגריידר לדוגמה מדפיס בנוסף (השורה נקראת משמאל לימין):

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