

פארקי מזרקות

בפארק אזורי, ישנן n **מזרקות**, ממוספרות מ-0 עד $n - 1$. נמדל את המזרקות כנקודות על מישור דו מימדי. המזרקה ה- i ($0 \leq i \leq n - 1$) היא נקודה $(x[i], y[i])$, כאשר $x[i]$ ו- $y[i]$ הם **שלמים זוגיים**. מיקומי המזרקות שונים זה מזה.

טימותי הארכיטקט אחראי לתכנון הבנייה של כמה **שבילים** ולמיקום של **ספסל** אחד לשביל. שביל הוא קטע ישר **אופקי** או **אנכי** באורך 2, שקצותיו הן שתי מזרקות שונות. השבילים צריכים להיבנות בצורה שניתן להגיע מכל מזרקה אחת לאחרת על ידי מעבר בשבילים. במצב ההתחלתי, אין שבילים בפארק.

לכל שביל, **בדיוק** ספסל אחד צריך להיות ממוקם בפארק **ומשוייך** לשביל זה, כלומר פונה אליו. כל ספסל צריך להיות ממוקם בנקודה כלשהי (a, b) כך ש- a ו- b הם **שלמים אי זוגיים**. המיקומים של הספסלים חייבים להיות **שונים** זה מזה. ספסל בנקודה (a, b) יכול להיות משוייך לשביל רק אם **שני** קצוות השביל הם מהבאים: $(a - 1, b - 1)$, $(a - 1, b + 1)$, $(a + 1, b - 1)$ ו- $(a + 1, b + 1)$. לדוגמה, הספסל בנקודה $(3, 3)$ יכול להיות משוייך רק לשביל, שהוא אחד מארבעת הקטעים הישרים הבאים: $(2, 2) - (2, 4)$, $(2, 4) - (4, 4)$, $(4, 4) - (4, 2)$, $(4, 2) - (2, 2)$.

עזרו לטימותי להחליט האם אפשרי לבנות שבילים, למקם ולשייך ספסלים בצורה המספקת את התנאים לעיל. אם כן, מסרו לו פתרון אפשרי. אם ישנם כמה פתרונות אפשריים המקיימים את התנאים, מסרו אחד מהם.

פרטי מימוש

יש לממש את הפונקציה הבאה:

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

- x, y : שני מערכים באורך n . לכל i ($0 \leq i \leq n - 1$), המזרקה ה- i ממוקמת בנקודה $(x[i], y[i])$, כש- $x[i]$ ו- $y[i]$ הם שלמים זוגיים.
- אם קיים פתרון אפשרי, הפונקציה צריכה לקרוא בדיוק פעם אחת ל-`build` (מפורט מטה) על מנת לדווח על פתרון, ולאחר מכן להחזיר 1.
- אחרת, הפונקציה צריכה להחזיר 0 מבלי לקרוא ל-`build`.
- הפונקציה נקראת פעם אחת בלבד.

אתם יכולים במימוש הפונקציה שלכם לקרוא לפונקציה הבאה, על מנת לספק בנייה אפשרית של שבילים ומיקום של ספסלים:

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

- יהי m מספר השבילים הכולל בבנייה.
- u, v : שני מערכים באורך m , המייצגים את השבילים המיועדים לבנייה. השבילים ממוספרים מ-0 עד $m - 1$.
- לכל j ($0 \leq j \leq m - 1$), שביל j מחבר את מזרקות $u[j]$ ו- $v[j]$. כל שביל הוא קטע ישר אופקי או אנכי

באורך 2. לכל שני שבילים שונים יש לכל היותר נקודה אחת משותפת (מזרקה). לאחר בניית השבילים, אמור להיות אפשרי להגיע מכל מזרקה לאחרת על ידי מעבר בהם.

- a, b : שני מערכים באורך m , המייצגים את הספסלים. לכל j ($0 \leq j \leq m - 1$), ממוקם ספסל בנקודה $(a[j], b[j])$, ומשוייך לשביל j . אסור שיהיו שני ספסלים שונים בעלי אותו מיקום.

דוגמאות

דוגמה 1

הקריאה הבאה:

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

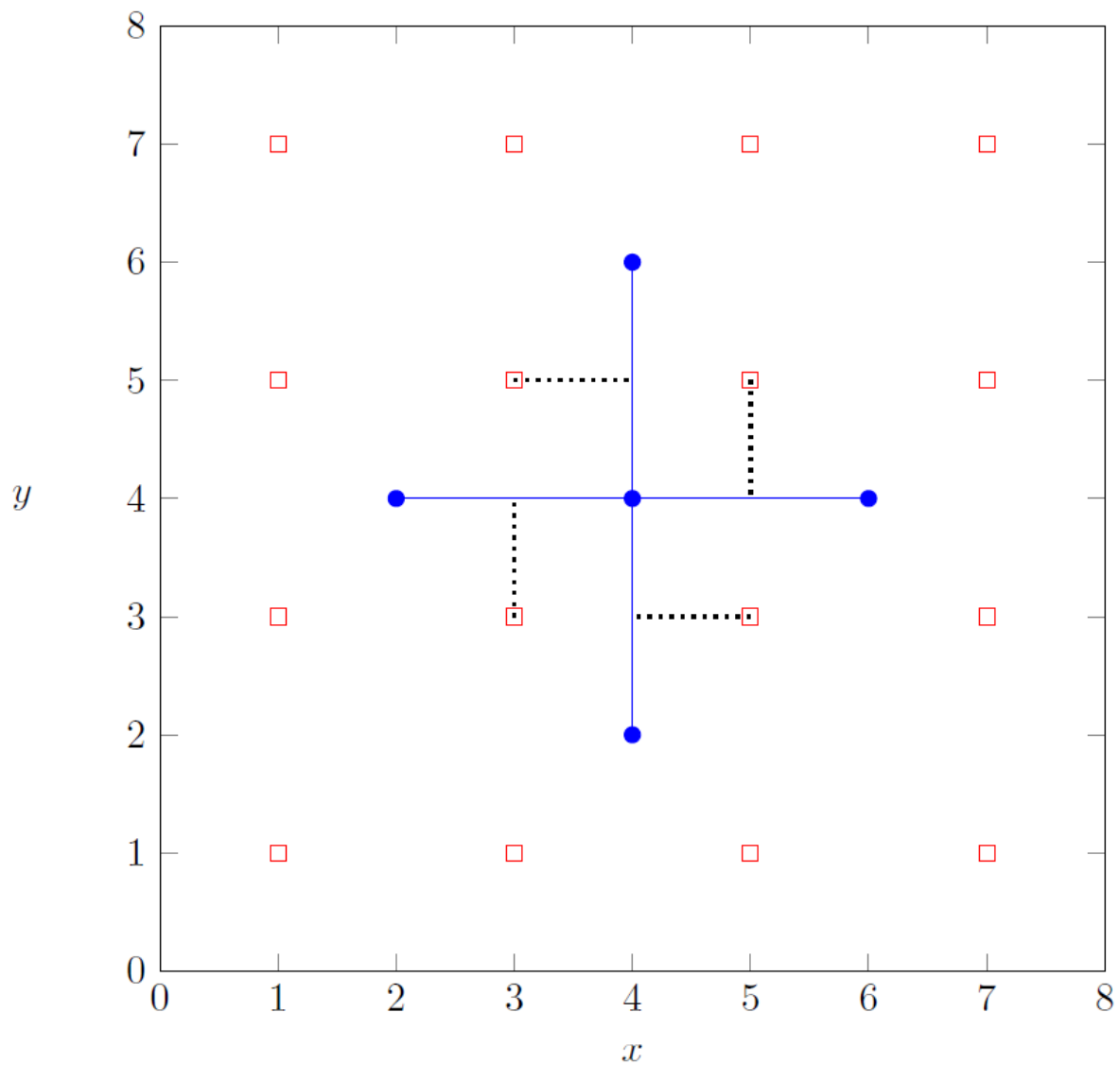
מציינת שישנן 5 מזרקות:

- מזרקה 0 ממוקמת בנקודה $(4, 4)$,
- מזרקה 1 ממוקמת בנקודה $(4, 6)$,
- מזרקה 2 ממוקמת בנקודה $(6, 4)$,
- מזרקה 3 ממוקמת בנקודה $(4, 2)$,
- מזרקה 4 ממוקמת בנקודה $(2, 4)$.

אפשרי לבנות את 4 השבילים הבאים, כך שכל שביל מחבר שתי מזרקות, ולמקם את הספסלים המשוייכים לכל שביל:

מספר שביל	מספרי המזרקות שהשביל מחבר	מיקום הספסל המשוייך לשביל
0	0, 2	(5, 5)
1	0, 1	(3, 5)
2	3, 0	(5, 3)
3	4, 0	(3, 3)

פתרון זה מתאים לסרטוט הבא:



כדי לדווח על פתרון זה, על `construct_roads` לבצע את הקריאה הבאה:

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

ולאחר מכן להחזיר 1.

שימו לב שבמקרה זה, ישנם מספר פתרונות המקיימים את הדרישות, וכולם יתקבלו. לדוגמה, גם הקריאה הבאה תתקבל:

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

ולאחר מכן להחזיר 1.

דוגמה 2

בקריאה הבאה:

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

מזרקה 0 ממוקמת בנקודה (2, 2) ומזרקה 1 ממוקמת בנקודה (4, 6). מאחר ואין דרך לבנות את השבילים ולקיים את התנאים, `construct_roads` תחזיר 0 מבלי לקרוא ל-`build`.

מגבלות

- $1 \leq n \leq 200\,000$
- $(0 \leq i \leq n - 1) \quad 2 \leq x[i], y[i] \leq 200\,000$ (לכל i)
- $x[i]$ ו- $y[i]$ הם שלמים זוגיים (לכל i , $0 \leq i \leq n - 1$).
- אין שתי מזרקות באותו המיקום.

תת משימות

1. (5 נקודות) $x[i] = 2$ (לכל i , $0 \leq i \leq n - 1$)
2. (10 נקודות) $2 \leq x[i] \leq 4$ (לכל i , $0 \leq i \leq n - 1$)
3. (15 נקודות) $2 \leq x[i] \leq 6$ (לכל i , $0 \leq i \leq n - 1$)
4. (20 נקודות) יש לכל היותר דרך אחת לבנות את השבילים, כך שניתן להגיע מכל מזרקה אחת לאחרת ע"י תנועה בשבילים.
5. (20 נקודות) לא קיימות ארבע מזרקות המהוות פינות של ריבוע 2×2 .
6. (30 נקודות) ללא מגבלות נוספות.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

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

הפלט של הגריידר לדוגמה הוא בפורמט הבא:

- שורה 1: ערך ההחזרה של `construct_roads`

אם ערך ההחזרה של `construct_roads` הוא 1 ו-`build(u, v, a, b)` נקראת, הגריידר מדפיס בנוסף:

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