



## Vision Program

הנכם מממשים מערכת ראייה עבור רובוט. בכל פעם שמצלמת הרובוט מצלמת תמונה, היא נשמרת בשחור-לבן בזיכרון שלו. כל תמונה היא טבלה של פיקסלים בגודל  $H \times W$ , עם שורות הממוספרות מ-0 עד  $H - 1$  ועמודות הממוספרות מ-0 עד  $W - 1$ . בכל תמונה יש **בדיוק שני** פיקסלים שחורים, ושאר הפיקסלים הם לבנים.

הרובוט יעבד את התמונה באמצעות תוכנית המורכבת מהוראות פשוטות. נתונים לכם  $W, H$ , וכן שלם חיובי  $K$ . עליכם לכתוב פונקציה שתייצר תוכנית עבור הרובוט. לכל תמונה, התוכנית צריכה לקבוע האם המרחק בין שני הפיקסלים הוא בדיוק  $K$ . נגדיר את המרחק בין הפיקסל שבשורה  $r_1$  ובעמודה  $c_1$  לבין הפיקסל שבשורה  $r_2$  ובעמודה  $c_2$  על ידי  $|r_1 - r_2| + |c_1 - c_2|$ . בנוסחה זו  $|x|$  מסמן את הערך המוחלט של  $x$ , השווה ל- $x$  אם  $x \geq 0$  ול- $-x$  אם  $x < 0$ .

כעת נסביר כיצד פועל הרובוט.

הזיכרון של הרובוט הוא מערך חד-מימדי מספיק גדול, שהתאים שלו ממוספרים החל מ-0. כל תא יכול להכיל 0 או 1, וערכו לא ישתנה לאחר שנקבע. התמונה מאוחסנת במערך שורה אחר שורה, בתאים הממוספרים מ-0 עד  $H \cdot W - 1$ . השורה הראשונה מאוחסנת בתאים 0 עד  $W - 1$ , והשורה האחרונה מאוחסנת בתאים  $(H - 1) \cdot W$  עד  $H \cdot W - 1$ . אם הפיקסל בשורה  $i$  ובעמודה  $j$  הוא שחור, אז הערך של התא  $i \cdot W + j$  הוא 1, ואחרת הוא 0.

התוכנית של הרובוט היא סדרה של **הוראות**, הממוספרות ברצף החל מ-0. בעת ריצת התוכנית ההוראות מבוצעות בזו אחר זו. לכל הוראה **קלטים** שהם תא אחד או יותר מזיכרון הרובוט, ו**פלט** שהוא ערך יחיד השווה ל-0 או ל-1. הפלט של הוראה מספר  $i$ , מאוחסן בתא  $H \cdot W + i$ . הקלטים של הוראה  $i$  חייבים להיות תאים המאחסנים פיקסלים או פלטים של הוראות קודמות. כלומר, הקלטים הם מהתאים 0 עד  $H \cdot W + i - 1$ .

יש ארבעה סוגי הוראות:

- NOT: מקבלת קלט אחד בדיוק. הפלט הוא 1 אם הקלט הוא 0, ואחרת הוא 0.
- AND: מקבלת קלט אחד או יותר. הפלט הוא 1 אם ורק אם **כל** הקלטים הם 1.
- OR: מקבלת קלט אחד או יותר. הפלט הוא 1 אם ורק אם **לפחות** אחד מהקלטים הוא 1.
- XOR: מקבלת קלט אחד או יותר. הפלט הוא 1 אם ורק אם מספר הקלטים השווים ל-1 הוא **אי-זוגי**.

הפלט של ההוראה האחרונה בתוכנית צריך להיות 1 אם המרחק בין שני הפיקסלים השחורים הוא בדיוק  $K$ , ואחרת 0.

## פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
void construct_network(int H, int W, int K)
```

- $H, W$ : מימדי התמונות שיצלם הרובוט
- $K$ : שלם חיובי
- הפונקציה צריכה לייצר את התוכנית של הרובוט. עבור כל תמונה שיקבל הרובוט, התוכנית שלו תקבע האם המרחק בין שני הפיקסלים השחורים בתמונה הוא בדיוק  $K$ .
- הפונקציה צריכה לקרוא לאחת או יותר מהפונקציות הבאות, כדי להוסיף הוראות לתוכנית של הרובוט (בהתחלה התוכנית ריקה):

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- הפונקציות מוסיפות את ההוראות NOT, AND, OR, או XOR, בהתאמה.
- $N$  (בפונקציה add\_not): האינדקס של הקלט להוראה NOT
- $Ns$  (בפונקציות add\_and, add\_or, ו-add\_xor): מערך של האינדקסים של הקלטים להוראה.
- כל אחת מהפונקציות מחזירה את האינדקס של התא בו יאוחסן הפלט של ההוראה. סדרת הקריאות לפונקציות אלו תחזיר סדרת מספרים עוקבים שמתחילה ב- $H \cdot W$ .

התוכנית של הרובוט יכולה להכיל לכל היותר 10 000 הוראות. ההוראות יכולות לקרוא לכל היותר 1 000 000 ערכים בסך הכל. כלומר, סכום אורכי  $Ns$  בכל הקריאות ל-add\_and, add\_or, ו-add\_xor בתוספת מספר הקריאות ל-add\_not יהיה לכל היותר 1 000 000.

לאחר הוספת ההוראה האחרונה, על הפונקציה construct\_network להסתיים. לאחר מכן התוכנית של הרובוט תיבדק על מספר תמונות. פתרונוכם ל-testcase מסוים יחשב נכון אם לכל אחת מתמונות אלו, הפלט של ההוראה האחרונה הוא 1 אם ורק אם המרחק בין שני הפיקסלים השחורים בתמונה הוא  $K$ .

הגריידר עשוי להדפיס אחת מהודעות השגיאה הבאות:

- Instruction with no inputs: התקבל מערך ריק כקלט ל-add\_and, ל-add\_or, או ל-add\_xor.
- Invalid index: אינדקס שגוי (שלילי למשל) ניתן כקלט ל-add\_and, ל-add\_or, או ל-add\_xor.
- Too many instructions: הפונקציה שלכם ניסתה להוסיף יותר מ-10 000 הוראות.
- Too many inputs: ההוראות קוראות יותר מ-1 000 000 ערכים בסך הכל.

## דוגמה

נניח  $H = 2, W = 3, K = 3$ . יש רק שתי תמונות אפשריות בהן המרחק בין הפיקסלים השחורים הוא 3.

0	1	2
3	4	5

0	1	2
3	4	5

- מקרה 1: הפיקסלים השחורים הם 0 ו-5
- מקרה 2: הפיקסלים השחורים הם 2 ו-3

פתרון אפשרי הוא לבנות תוכנית לרובוט בעזרת הקריאות הבאות:

1.  $\text{add\_and}([0, 5])$ , מוסיפה הוראה הפולטת 1 אם ורק אם קורה המקרה הראשון. הפלט מאוחסן בתא 6.
2.  $\text{add\_and}([2, 3])$ , מוסיפה הוראה הפולטת 1 אם ורק אם קורה המקרה השני. הפלט מאוחסן בתא 7.
3.  $\text{add\_or}([6, 7])$ , מוסיפה הוראה הפולטת 1 אם ורק אם אחד המקרים לעיל.

## מגבלות

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

## תת-משימות

1.  $\max(H, W) \leq 3$  (10 נקודות)
2.  $\max(H, W) \leq 10$  (11 נקודות)
3.  $\max(H, W) \leq 30$  (11 נקודות)
4.  $\max(H, W) \leq 100$  (15 נקודות)
5.  $\min(H, W) = 1$  (12 נקודות)
6. (8 נקודות) הפיקסל שבשורה 0 ועמודה 0 הוא שחור בכל תמונה.
7. (14 נקודות)  $K = 1$
8. (19 נקודות) ללא מגבלות נוספות.

## גריידר לדוגמה

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

- שורה 1:  $H \ W \ K$
- שורה  $2 + i$  ( $i \geq 0$ ):  $r_1[i] \ c_1[i] \ r_2[i] \ c_2[i]$
- שורה אחרונה: -1

כל שורה פרט לראשונה ולאחרונה מייצגת תמונה יחידה. שימו לב: תמונה  $i$  ( $i \geq 0$ ) מתוארת בשורה  $i + 2$ . פיקסל שחור אחד נמצא בשורה  $r_1[i]$  ובעמודה  $c_1[i]$  והאחר נמצא בשורה  $r_2[i]$  ובעמודה  $c_2[i]$ .

הגריידר לדוגמה קורא ל-`construct_network(H, W, K)`. אם `construct_network` מפרה חלק מהמגבלות המתוארות בשאלה, הגריידר לדוגמה מדפיס אחת מהודעות השגיאה המתוארות בסוף החלק "פרטי מימוש" ומסיים את ריצתו.

אחרת, הגריידר לדוגמה מייצר שני פלטים.

ראשית, הגריידר לדוגמה מדפיס את הפלט של תוכנית הרובוט בפורמט הבא:

• שורה  $i + 1$  ( $0 \leq i$ ): פלט ההוראה של תוכנית הרובוט עבור תמונה  $i$  (1 או 0).

שנית, הגריידר לדוגמה כותב לקובץ `log.txt` בתיקייה הנוכחית בפורמט הבא:

• שורה  $i + 1$  ( $0 \leq i$ ):  $m[i][0] \ m[i][1] \ \dots \ m[i][c - 1]$

בשורה  $i + 1$  רשומים הערכים הנמצאים בזיכרון הרובוט בסיום ריצת תוכנית על תמונה  $i$ . כלומר,  $m[i][j]$  הוא הערך של תא  $j$ . שימו לב שהערך של  $c$  (אורך השורה) שווה ל- $H \cdot W$  ועוד מספר ההוראות בתוכנית של הרובוט.