



Görmə proqramı

Robot üçün görmə proqramı hazırlayırsınız. Hər dəfə robotun kamerası şəkil çəkəndə, bu şəkil **robotun yaddaşında** ağ-qara şəkil kimi saxlanır. Hər bir şəkil $H \times W$ düzbucaqlı şəkildə piksellərdən ibarətdir. Sətirlər 0 dan $H - 1$ ə və sütunlar 0 dan $W - 1$ ə nömrələnmişdir. Hər bir şəkildə **tam olaraq iki** qara piksel var və qalan bütün piksellər ağdır.

Robot hər bir şəkli sadə əmrlərdən ibarət proqram vasitəsilə emal edə bilər. Sizə H , W və müsbət tam ədəd K verilir. Sizin məqsədiniz robot proqramına istənilən şəkil üçün iki qara piksel arasındakı **məsafənin** dəqiq K olub olmadığını müəyyən edən prosedür yazmaqdır. Burada r_1 sətir, c_1 sütunundakı və r_2 sətir, c_2 sütunundakı piksellər arasındakı məsafə $|r_1 - r_2| + |c_1 - c_2|$ dir. Bu düsturda $|x|$ x in mütləq qiymətini göstərir.

İndi robotun necə işlədiyini təsvir edəcəyik.

Robotun yaddaşı kifayət qədər böyük massiv şəkildə xanalardan ibarətdir. Xanalar 0 dan nömrələnir. Hər bir xanada 0 və ya 1 saxlanıla bilər və hər hansı biri xanaya yazıldığı zaman, bu xana daha dəyişdirilə bilməz. Şəkil 0 dan $H \cdot W - 1$ ə qədər olan xanalarda sətir-sətir saxlanılır. İlk sətir 0 dan $W - 1$ ə, son sətir $(H - 1) \cdot W$ dan $H \cdot W - 1$ ə olan xanalarda saxlanılır. Xüsusilə, əgər i sətir və j sütundakı xana qara olarsa $i \cdot W + j$ xanasının dəyəri 1, əks halda 0 olur.

Robotun proqramı 0 dan başlayaraq nömrələnmiş **əmrilər** ardıcılığından ibarətdir. Proqram çalışdığı zaman əmrilər bir-bir icra olunur. Hər bir əmr bir və ya bir neçə xananın dəyərini oxuyur (biz bu dəyərləri əmrin **girişi** adlandırırıq) və bir dəyər $- 0$ və ya 1 emal edir (biz bu dəyəri əmrin **çıxışı** adlandırırıq). i -ci əmrin çıxışı $H \cdot W + i$ xanasında saxlanır. i -ci əmrin girişi ya pikselləri saxlayan xanalar və ya əvvəlki əmrlərin çıxışı ola bilər, yəni, 0 dan $H \cdot W + i - 1$ ə olan xanalar.

Dörd cür əmr var:

- NOT: bir girişi var. Giriş 0 olduğu zaman çıxış 1, əks halda çıxış 0 olur.
- AND: bir və ya daha çox girişi ola bilər. **Bütün** girişlər 1 olduğu zaman çıxış 1, əks halda çıxış 0 olur.
- OR: bir və ya daha çox girişi ola bilər. **Ən azı bir** giriş 1 olduğu zaman çıxış 1, əks halda çıxış 0 olur.
- XOR: bir və ya daha çox girişi ola bilər. **Tək sayda** giriş 1 olduğu zaman çıxış 1, əks halda çıxış 0 olur.

İki qara piksel arasındakı məsafə tam olaraq K olduğu halda ən son əmrin çıxışı 1, əks halda 0 olmalıdır.

İmplementasiya detalları

Aşağıdakı proseduru realizə (implement) etməlisiniz:

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

- H, W : robotun kamerasından çəkilən şəklın ölçüləri
- K : tam ədəd
- Bu prosedur robotun proqramını emal etməlidir. Bu proqram robotun kamerasından çəkilən istənilən şəkil üçün iki qara piksel arasındakı məsafənin tam olaraq K olub olmadığını müəyyən etməlidir.

Bu prosedur aşağıdakı prosedurlardan bir və ya bir neçəsini robotun proqramına (başlanğıcda boşdur) artırmaq üçün çağırılmalıdır:

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

- Uyğun olaraq NOT, AND, OR və ya XOR əmrini əlavə etmək.
- N (`add_not` üçün): NOT əmrinin girişini oxuduğu xananın nömrəsi.
- Ns (`add_and`, `add_or`, `add_xor` üçün): AND, OR və ya XOR əmrinin girişini oxuduğu xanaların nömrələrindən ibarət massiv.
- Hər bir prosedur əmrin çıxışını saxlayan xananın nömrəsini qaytarır. Bu prosedurların ardıcıl çağırılması $H \cdot W$ dən başlayan ardıcıl nömrələr qaytarır.

The robot's program can consist of at most 10 000 instructions. The instructions can read at most 1 000 000 values in total. In other words, the total length of Ns arrays in all calls to `add_and`, `add_or` and `add_xor` plus the number of calls to `add_not` cannot exceed 1 000 000.

After appending the last instruction, procedure `construct_network` should return. The robot's program will then be evaluated on some number of images. Your solution passes a given test case if for each of these images, the output of the last instruction is 1 if and only if the distance between the two black pixels in the image is equal to K .

The grading of your solution may result in some of the messages in English, which are explained below:

- Instruction with no inputs: an empty array was given as the input to `add_and`,

add_or, or add_xor.

- Invalid index: an incorrect (possibly negative) cell index was provided as the input to add_and, add_or, add_xor, or add_not.
- Too many instructions: your procedure attempted to add more than 10 000 instructions.
- Too many inputs: the instructions read more than 1 000 000 values in total.

Example

Assume $H = 2$, $W = 3$, $K = 3$. There are only two possible images where the distance between the black pixels is 3.

0	1	2
3	4	5

0	1	2
3	4	5

- Case 1: black pixels are 0 and 5
- Case 2: black pixels are 2 and 3

A possible solution is to build a robot's program by making the following calls:

1. add_and([0, 5]), which adds an instruction that outputs 1 if and only if the first case holds. The output is stored in cell 6.
2. add_and([2, 3]), which adds an instruction that outputs 1 if and only if the second case holds. The output is stored in cell 7.
3. add_or([6, 7]), which adds an instruction that outputs 1 if and only if one of the cases above holds.

Constraints

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

Subtasks

1. (10 points) $\max(H, W) \leq 3$
2. (11 points) $\max(H, W) \leq 10$
3. (11 points) $\max(H, W) \leq 30$
4. (15 points) $\max(H, W) \leq 100$
5. (12 points) $\min(H, W) = 1$
6. (8 points) Pixel in row 0 and column 0 is black in each image.

7. (14 points) $K = 1$
8. (19 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

- line 1: $H W K$
- line $2 + i$ ($i \geq 0$): $r_1 c_1 r_2 c_2$
- last line: -1

Each line excepting the first and the last line represents an image with two black pixels. We denote the image described on line $2 + i$ by image i . One black pixel is in row r_1 and column c_1 and the other one in row r_2 and column c_2 .

The sample grader may print `Invalid user input` if it detects any error in the input (e.g., the input is referencing a non-existing row or column).

If no error is detected, the sample grader prints the output of the robot's program in the following format:

- line $1 + i$ ($0 \leq i$): output of the last instruction in the robot's program for image i (1 or 0).

In addition, the sample grader writes a file `log.txt` in the current directory in the following format:

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

The sequence on line $1 + i$ describes the values stored in the robot's memory cells after the robot's program is run, given image i as the input. Specifically, $m[j]$ gives the value of cell j . Note that the value of c (the length of the sequence) is equal to $H \cdot W$ plus the number of instructions in the robot's program.