



Képertékelő program

Képek értékelését végző robotprogramot kell megvalósítanod. Amikor a robot egy képet kap, azt fekete-fehér képként tárolja a memóriájában. Minden kép H -szor W képpontból álló rács, ahol a sorok 0-tól $H - 1$ -ig és az oszlopok 0-tól $W - 1$ -ig sorszámozódnak. Minden képen **pontosan kettő** fekete képpont van, minden más képpont fehér.

A robot minden képre egyszerű utasításokból álló programot hajt végre. Adott H , W és egy pozitív egész K . A robot számára olyan programot kell készíteni, amely minden képre meghatározza, hogy a két fekete képpont **távolsága** pontosan K -e. Az r_1 sorban és a c_1 oszlopban valamint az r_2 sorban és c_2 oszlopban található pontok távolsága: $|r_1 - r_2| + |c_1 - c_2|$. Ebben a képletben az $|x|$ az x abszolút értékét jelenti, ami x , ha $x \geq 0$ és $-x$, ha $x < 0$.

A robot az alábbiak szerint működik.

A robotnak elegendően nagy a memóriája, a memóriacellák 0-tól indexelődnek. Minden memóriacella a 0 vagy az 1 értéket tartalmazza, az értéke csak egyszer állítható be és utána nem változhat. A kép sorfolytonosan tárolódik a memóriacellákban, 0-tól a $H \cdot W - 1$. memóriacelláig. Az első sor $0, \dots, W - 1$ memóriacellákban van tárolva, az utolsó sor a $(H - 1) \cdot W, \dots, H \cdot W - 1$ memóriacellákban. Azaz, ha az i . sor és j . oszlopban levő képpont fekete, akkor az $i \cdot W + j$ memóriacella értéke 1, különben ez az érték 0.

A robot programja **utasítások** sorozatából áll, amelyeket 0-tól sorszámozunk. A program az utasításokat sorról-sorra, egyenként hajtja végre. Minden egyes utasítás egy vagy több cella értékét olvassa (ezeket az utasítás **input**-jainak nevezzük). Az utasítás végrehajtása egyetlen, 0 vagy 1 értéket eredményez (ez az érték az utasítás **output**-ja). Az i . utasítás outputja a $H \cdot W + i$. memóriacellában tárolódik el. Az i . utasításnak csak olyan cellák lehetnek az inputjai, amik képpontok vagy egy korábbi utasítás outputjai, vagyis a $0, \dots, H \cdot W + i - 1$ memóriacellák értékei.

Négyféle utasítás van:

- NOT: Pontosán egyetlen inputja van. Az output 1, ha az input 0, egyébként az output 0.
- AND: Egy vagy több bemenete lehet. Az output akkor és csak akkor 1, ha **minden** input értéke 1.
- OR: Egy vagy több bemenete lehet. Az output akkor és csak akkor 1, ha **legalább egy** input értéke 1.

- XOR: Egy vagy több bemenete lehet. Az output akkor és csak akkor 1, ha az inputban az 1 értékek száma **páratlan**.

A program utolsó utasításának az outputja legyen 1, ha a két fekete képpont távolsága pontosan K , egyébként legyen 0.

Megvalósítás

Az alábbi függvényt kell megvalósítanod:

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

- H, W : a robot kamerája által rögzített képek dimenziói
- K : pozitív egész
- A függvénynek a robot programját kell legenerálnia. Ennek a robotprogramnak bármely, a robot kamerája által készített képre meg kell tudnia állapítani, hogy a két fekete képpont távolsága pontosan K -e.

z függvény az alábbi függvényhívásokkal tud utasítást hozzáfűzni az elkészítendő robotprogramhoz (ami kezdetben üres):

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

- A végére fűzi sorban a NOT, az AND, az OR vagy az XOR utasítást.
- N (add_not): annak a cellának a sorszáma, amely a NOT utasítás inputja.
- Ns (add_and, add_or, add_xor): azon cellák indexeit tartalmazó tömb, amelyek az AND, az OR vagy az XOR utasítások inputjai.
- Mindegyik függvény az output cella sorszámát adja visszatérési értékül. A függvények egymás utáni hívása egymást követő egész számokat ad $H \cdot W$ -tól kezdődően.

A robotprogram legfeljebb 10 000 utasításból állhat. Az utasítások legfeljebb 1 000 000 értéket olvashatnak. Más szóval, az összes Ns tömbök elemszáma az összes add_and, add_or és add_xor hívásban és az add_not hívások száma nem lehet több, mint 1 000 000.

Az utolsó utasítás hívása után a construct_network végrehajtásának be kell fejeződnie. A robotprogramot néhány képre, mint bemenetre végrehajtva tesztelik. A megoldás egy tesztesetre akkor kap pontot, ha a tesztesethez tartozó minden képbemenetre helyes eredményt ad.

Az értékelő az alábbi angol nyelvű üzeneteket adhatja:

- Instruction with no inputs: az `add_and`, `add_or` vagy `add_xor` utasításnak üres input paramétere volt.
- Invalid index: érvénytelen (esetleg negatív) cellahivatkozás kapott az `add_and`, `add_or`, `add_xor` vagy `add_not` utasítás.
- Too many instructions: a függvény több, mint 10 000 utasítást akar hozzáadni a robotprogramhoz.
- Too many inputs: az összes input elemek száma több, mint 1 000 000.

Példa

Legyen $H = 2$, $W = 3$, $K = 3$. Csak két olyan kép lehetséges, amelyben a fekete képpontok távolsága 3.

0	1	2
3	4	5

0	1	2
3	4	5

1. eset: A fekete pontok a 0. és az 5.
2. eset: A fekete pontok a 2. és a 3.

Egy lehetséges robotprogram a következő három utasításból áll:

1. `add_and([0, 5])`, ennek az utasításnak az outputja akkor és csak akkor 1, ha a kép az 1. esetnek felel meg. Az output a 6. cellába kerül.
2. `add_and([2, 3])`, ennek az utasításnak az outputja akkor és csak akkor 1, ha a kép az 2. esetnek felel meg. Az output a 7. cellába kerül.
3. `add_or([6, 7])`, ennek az utasításnak az outputja akkor és csak akkor 1, ha a felsorolt két eset valamelyike áll fenn.

Korlátok

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

Pontozás

1. (10 pont) $\max(H, W) \leq 3$
2. (11 pont) $\max(H, W) \leq 10$
3. (11 pont) $\max(H, W) \leq 30$
4. (15 pont) $\max(H, W) \leq 100$
5. (12 pont) $\min(H, W) = 1$

6. (8 pont) 0. sorban és 0. oszlopban levő képpont minden képen fekete.
7. (14 pont) $K = 1$
8. (19 pont) Nincs egyéb feltétel.

Mintaértékelő

A mintaértékelő az alábbi formában olvassa a bemenetet:

- 1. sor: $H \ W \ K$
- $2 + i$. sor: ($i \geq 0$): $r_1 \ c_1 \ r_2 \ c_2$
- utolsó sor: -1

Az első és az utolsó sor kivételével minden sor egy képet ad meg, pontosan két fekete pont koordinátaival. Az i . kép a $2 + i$. sorban van. Az egyik fekete pont az r_1 . sor c_1 . oszlopában, míg a másik az r_2 . sor c_2 . oszlopában van.

A mintaértékelő az `Invalid user input` üzenetet adja, ha az input nem megfelelő (például ha az inputban nemlétező sor vagy oszlop van).

Hibátlan futás esetén a mintaértékelő a robotprogram outputját írja ki az alábbi formában:

- $1 + i$. sor ($0 \leq i$): a robotprogram utolsó utasításának outputja az i . képre (1 vagy 0).

A mintaértékelő egy `log.txt` fájlt is készít az aktuális mappába, amelynek a formája a következő:

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

Az $1 + i$. sorban az i . képhez tartozó, a robotprogram lefutása utáni robotmemória értékei találhatóak. Az $m[j]$ a j . cella értéke. A c értéke, a sorozat elemeinek a száma: $H \cdot W +$ (a robotprogram utasításainak a száma).