Roboto regėjimo programa

Kuriate programą, kuri realizuoja roboto regėjimą. Kiekvieną kartą, kai roboto kamera nufotografuoja vaizdą, jis išsaugomas roboto atmintyje kaip juodai baltas paveikslėlis. Kiekvienas paveikslėlis yra $H \times W$ pikselių tinklelis, kurio eilutės sunumeruotos nuo 0 iki $H - 1$, stulpeliai — nuo 0 iki $W - 1$. Kiekviename paveikslėlyje yra **lygiai du** juodi pikseliai, o visi likusieji yra balti.

Kiekvieną paveikslėlį robotas apdoroja naudodamasis programa, kurią sudaro paprastos komandos. Duotos H ir W reikšmės bei teigiamas sveikasis skaičius K. Parašykite procedūrą, kuri sugeneruotų robotui skirtą programą. Sugeneruotoji programa turi bet kuriam paveikslėliui nustatyti, ar **atstumas** tarp dviejų juodų pikselių yra lygiai K. Atstumas tarp pikselio, esančio eilutėje r_1 ir stulpelyje c_1, ir pikselio, esančio eilutėje r_2 ir stulpelyje c_2, lygus $|r_1 - r_2| + |c_1 - c_2|$. Šioje formulėje $|x|$ žymi x modulį, kuris lygus x, jei $x \geq 0$ ir lygus $-x$, jei $x < 0$.

Toliau aprašysime, kaip veikia robotas.

Roboto atmintį sudaro pakankamai didelis masyvas, kurio elementai sunumeruoti nuo 0. Į kiekvieną atminties laukelį galima įrašyti 0 arba 1 ir vieną kartą įrašius reikšmę, ji nebegali būti keičiama. Paveikslėlis saugomas eilutė po eilutės atminties laukeliuose, kurių numeriai yra nuo 0 iki $H \cdot W - 1$. Pirmoji eilutė saugoma nuo 0-inio iki $(W - 1)$-ojo atminties laukeliuose, o paskutinioji eilutė saugoma atminties laukeliuose nuo $((H - 1) \cdot W)$-ojo iki $(H \cdot W - 1)$-ojo. Jeigu pikselis, esantis eilutėje i ir stulpelyje j, yra juodas, $(i \cdot W + j)$-ojo laukelio reikšmė lygi 1, kitu atveju — 0.

Robotui skirtą programą sudaro **komandų**, sunumeruotų iš eilės, pradedant 0, seka. Vykdydamas programą, robotas nuosekliai po vieną vykdo komandas. Kiekviena komanda perskaito vienos ar daugiau laukelių reikšmes (tas reikšmes vadinsime komandos **argumentais**) ir suskaičiuoja vieną reikšmę, lygią 0 arba 1 (šią reikšmę vadinsime komandos **rezultatu**). Komandos i rezultatas išsaugomas laukelyje $H \cdot W + i$. Komandos i argumentai gali būti tik tie laukeliai, kuriuose saugomos pikselių reikšmės arba prieš tai buvusių komandų rezultatai, t.y. laukeliai, kurių numeriai nuo 0 iki $H \cdot W + i - 1$.

Komandos gali būti keturių tipų:

- `NOT`: turi lygiai vieną argumentą. Jos rezultatas yra 1, jei argumentas yra 0, kitu atveju rezultatas lygus 0.
- `AND`: turi vieną arba daugiau argumentų. Jos rezultatas lygus 1 tada ir tik tada, kai **visi** argumentai lygūs 1.

- `OR`: turi vieną arba daugiau argumentų. Jos rezultatas lygus 1 tada ir tik tada, kai **bent vienas** argumentas lygus 1.
- `XOR`: turi vieną arba daugiau argumentų. Jos rezultatas lygus 1 tada ir tik tada, kai argumentų, lygių 1, **skaičius yra nelyginis**.

Paskutiniosios komandos rezultatas turi būti lygus 1, jei atstumas tarp dviejų juodų pikselių lygus K, arba 0 priešingu atveju.

Realizacija

Parašykite šią procedūrą:

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

- H, W: kiekvieno roboto nufotografuoto paveikslėlio išmatavimai
- K: teigiamas sveikasis skaičius
- Ši procedūra turi sugeneruoti programą robotui. Kiekvienam roboto nufotografuotam paveikslėliui programa turi nustatyti, ar atstumas tarp abiejų juodų paveikslėlio pikselių lygus K.

Ši procedūra turi iškviesti vieną ar daugiau žemiau pateiktų procedūrų tam, kad jos būtų įtrauktos į roboto programą (kuri pradiniu momentu yra tuščia):

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

- Pridėti nurodytą komandą `NOT`, `AND`, `OR`, arba `XOR` .
- N (skirta `add_not`): laukelio, iš kurio pridedama `NOT` komanda skaitys argumentą, numeris
- Ns (skirta `add_and`, `add_or`, `add_xor`): masyvas su laukelių, iš kurių pridedama `AND`, `OR`, arba `XOR` komanda skaitys argumentus, numeriais
- Kiekviena procedūra grąžina laukelio, kuriame bus įrašytas rezultatas, numerį. Nuosekliai kviečiant šias procedūras, bus grąžinami paeiliui einantys sveikieji skaičiai, pradedant $H \cdot W$.

Roboto programą turi sudaryti ne daugiau nei $10\,000$ komandų. Visoms komandoms kartu leidžiama perskaityti ne daugiau $1\,000\,000$ reikšmių. Kitaip sakant, bendras visų Ns masyvų, kviečiant `add_and`, `add_or` ir `add_xor`, ilgis, kartu su kreipinių į `add_not` skaičiumi, negali viršyti $1\,000\,000$.

Pridėjus paskutiniąją komandą, procedūra `construct_network` turi baigti darbą. Robotui sugeneruota programa bus vertinama su tam tikru paveikslėlių skaičiumi. Jūsų sprendimas įveikia duotą testą, jei kiekvienam duotam paveikslėliui paskutiniosios

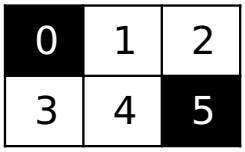
komandos rezultatas yra 1 tik jei atstumas tarp abiejų juodų pikselių lygus K.

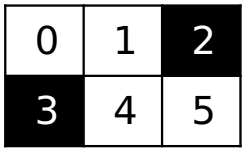
Vertinant sprendimą gali būti pateiktas vienas iš šių klaidos pranešimų:

- `Instruction with no inputs`: tuščias masyvas pateiktas kaip procedūros `add_and`, `add_or`, arba `add_xor` argumentas.
- `Invalid index`: nekorektiškas (galbūt neigiamas) laukelio numeris pateiktas kaip argumentas kviečiant procedūrą `add_and`, `add_or`, `add_xor`, arba `add_not`.
- `Too many instructions`: procedūra pabandė pridėti daugiau nei $10\,000$ komandų.
- `Too many inputs`: visos komandos kartu skaito daugiau nei $1\,000\,000$ reikšmių.

Pavyzdys

Tarkime, kad $H = 2$, $W = 3$, $K = 3$. Galimi tik du paveikslėliai, kuomet atstumas tarp juodų pikselių lygus 3.





- Pirmas atvejis: 0 ir 5 yra juodi pikseliai
- Antras atvejis: 2 ir 3 yra juodi pikseliai

Sugeneruoti programą robotui galite atlikdami šiuos kvietimus:

1. `add_and([0, 5])`, prideda komandą, kuri išveda 1 jei galioja pirmas atvejis. Rezultatas išsaugomas laukelyje 6.
2. `add_and([2, 3])`, prideda komandą kuri išveda 1 jei galioja antras atvejis. Rezultatas išsaugomas laukelyje 7.
3. `add_or([6, 7])`, prideda komandą kuri išveda 1 jei galioja lygiai vienas iš aukščiau pateiktų dviejų atvejų.

Ribojimai

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

Dalinės užduotys

1. (10 taškų) $\max(H, W) \leq 3$
2. (11 taškų) $\max(H, W) \leq 10$

3. (11 taškų) $\max(H, W) \leq 30$
4. (15 taškų) $\max(H, W) \leq 100$
5. (12 taškų) $\min(H, W) = 1$
6. (8 taškai) Kiekviename paveikslėlyje pikselis, esantis eilutėje 0 ir stulpelyje 0, yra juodas.
7. (14 taškų) $K = 1$
8. (19 taškų) Papildomų ribojimų nėra.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito duomenis tokiu formatu:

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

Kiekviena eilutė, išskyrus pirmąją ir paskutiniąją, aprašo paveikslėlį, kurio lygiai du pikseliai yra juodi. Laikykime, kad paveikslėlio, aprašyto $(2+i)$-oje eilutėje, numeris yra i. Vienas juodas pikselis yra eilutėje $r_1[i]$ ir stulpelyje $c_1[i]$, o kitas — eilutėje $r_2[i]$ ir stulpelyje $c_2[i]$.

Pavyzdinė vertinimo programa pirmiausia iškviečia `construct_network(H, W, K)`. Jei `construct_network` pažeidžia kurią nors sąlygą, nurodytą užduoties sąlygoje, pavyzdinė vertinimo programa išveda vieną iš anksčiau aprašytų klaidos pranešimų ir baigia darbą.

Priešingu atveju, pavyzdinė vertinimo programa pirmiausia išveda roboto programos rezultatą tokiu formatu:

- $(1+i)$-oji eilutė ($0 \leq i$): roboto paskutinės komandos rezultatas (1 arba 0) paveikslėliui i,

po to einamajame kataloge sukuria failą `log.txt` ir į jį rašo tokiu formatu:

- $(1+i)$-oji eilutė ($0 \leq i$): $m[i][0]$ $m[i][1]$ $\ldots$ $m[i][c-1]$

$(1+i)$-oje eilutėje pateikta seka aprašo roboto atminties laukeliuose įrašytas reikšmes, įvykdžius roboto programą su paveikslėliu i. Konkrečiau, $m[i][j]$ pateikia laukelio j reikšmę. Atkreipkite dėmesį, kad sekos ilgio c reikšmė lygi $H \cdot W$ ir roboto programoje esančių komandų skaičiaus sumai.