



Nägemisprogramm

Sa kirjutad robotile nägemisprogrammi. Kui robot teeb oma kaameraga pildi, salvestatakse see roboti mällu must-valge kujutisena. Iga selline kujutis koosneb $H \times W$ pikslist, mille read on nummerdatud 0 kuni $H - 1$ ja veerud 0 kuni $W - 1$. Igal kujutisel on **täpselt kaks** musta pikslit; ülejäänud pikslid on valged.

Robot töötleb iga kujutist talle ette antud programmi alusel. Sulle antakse H ja W väärtused ja lisaks täisarv K . Sinu ülesanne on kirjutada funktsioon, mis koostab robotile programmi, mis kontrollib iga võimaliku kujutise kohta, kas sellel oleva kahe musta piksli vaheline **kaugus** on täpselt K . Selles ülesandes loeme reas r_1 veerus c_1 oleva ja reas r_2 veerus c_2 oleva piksli vaheliseks kauguseks $|r_1 - r_2| + |c_1 - c_2|$. Avaldis $|x|$ tähistab x absoluutväärtust, mis on x , kui $x \geq 0$, ja $-x$, kui $x < 0$.

Nüüd kirjeldame roboti töötamist täpsemalt.

Roboti mälu on piisavalt suur massiiv, mille pesad on nummerdatud alates nullist. Igas mälupesas võib olla 0 või 1 ja seda ei saa pärast väärtuse omistamist enam muuta. Kujutis salvestatakse ridade kaupa pesadesse 0 kuni $H \cdot W - 1$. Kujutise esimene rida salvestatakse pesadesse 0 kuni $W - 1$ ja viimane rida pesadesse $(H - 1) \cdot W$ kuni $H \cdot W - 1$. Kui reas i veerus j olev piksel on must, siis on pesa $i \cdot W + j$ väärtus 1, muidu 0.

Roboti programm on **käskude** jada, mille elemendid on nummerdatud alates nullist. Programmi täitmisel täidetakse kõik käsud järjest ja ükshaaval. Iga käsk loeb ühe või mitme mälupesa sisu (mida me nimetame käsu **sisenditeks**) ja annab tulemuseks ühe väärtuse, mis võib olla 0 või 1 (ja mida me nimetame käsu **väljundiks**). Käsu number i väljund salvestatakse mälupesasse number $H \cdot W + i$. Käsu number i sisendid võivad olla mälupesad, mis sisaldavad kas kujutise piksleid või eelmiste käskude väljundeid, s.t. pesad 0 kuni $H \cdot W + i - 1$.

Käske on nelja liiki:

- NOT-käsul on täpselt üks sisend; väljund on 1 parajasti siis, kui sisend on 0.
- AND-käsul on üks või mitu sisendit; väljund on 1 parajasti siis, kui **kõik** sisendid on 1.
- OR-käsul on üks või mitu sisendit; väljund on 1 parajasti siis, kui **vähemalt üks** sisend on 1.
- XOR-käsul on üks või mitu sisendit; väljund on 1 parajasti siis, kui **paaritu arv** sisendeid on 1.

Programmi viimase käsu väljund peab olema 1, kui kujutise mustade pikslite vaheline kaugus on täpselt K , ja 0, kui see nii ei ole.

Realisatsioon

Lahendusena tuleb realiseerida järgmine funktsioon:

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

- H, W : roboti kaamera salvestatavate kujutiste kõrgus ja laius.
- K : positiivne täisarv.
- See funktsioon peab koostama robotile programmi, mis kontrollib iga võimaliku kujutise kohta, kas sellel oleva kahe musta piksli vaheline kaugus on täpselt K .

See funktsioon peab roboti (algselt tühjale) programmile käskude lisamiseks kutsuma ühe või rohkem kordi välja järgmisi funktsioone:

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

- Lisab programmi lõppu vastavalt NOT-, AND-, OR- või XOR-käsu.
- N (funktsioonis `add_not`): selle mälupeesa number, millest lisatav NOT-käsk oma sisendi loeb.
- Ns (funktsioonides `add_and`, `add_or`, `add_xor`): nende mälupejade numbrid, millest lisatav AND-, OR- või XOR-käsk oma sisendid loeb.
- Iga funktsioon tagastab selle mälupeesa numbri, kuhu lisatud käsk oma väljundi salvestab. Nende funktsioonide järjestikused väljakutsed tagastavad järjestikused täisarvud alates väärtusest $H \cdot W$.

Roboti programmis võib olla maksimaalselt 10 000 käsku. Need käsud võivad kokku lugeda maksimaalselt 1 000 000 väärtust. Teisisõnu, massiivide Ns pikkuste summa funktsioonide `add_and`, `add_or` ja `add_xor` kõigis väljakutsetes kokku pluss funktsiooni `add_not` väljakutsete arv ei tohi ületada 1 000 000.

Viimase käsu programmile lisamise järel peab funktsioon `construct_network` oma töö lõpetama. Seejärel kontrollitakse koostatud programmi õigsust mitme kujutisega. Sinu lahendus läbib testi edukalt, kui selle koostatud programm annab iga kujutise jaoks õige tulemuse (selle viimase käsu väljund on 1 parajasti siis, kui kahe musta piksli vaheline kaugus on K).

Sinu lahenduse hindamisel võib hindamisprogramm väljastada järgmisi teateid:

- `Instruction with no inputs`: sinu lahendus andis funktsioonile `add_and`,

add_or või add_xor sisendiks tühja massiivi.

- Invalid index: sinu lahendus andis funktsioonile add_and, add_or, add_xor või add_not sisendiks lubamatu (võimalik, et negatiivse) mälupeesa numbri.
- Too many instructions: sinu lahendus püüdis lisada programmi rohkem kui 10 000 käsku.
- Too many inputs: roboti programmile lisatud käsud püüdsid sisendina lugeda rohkem 1 000 000 väärtust.

Näide

Olgu $H = 2$, $W = 3$, $K = 3$. Siis leidub ainult kaks sellist kujutist, kus kahe musta piksli vaheline kaugus on 3.

0	1	2
3	4	5

0	1	2
3	4	5

- Variant 1: Mustad pikslid on 0 ja 5.
- Variant 2: Mustad pikslid on 2 ja 3.

Üks võimalik viis robotile programmi koostamiseks on järgmine:

1. `add_and([0, 5])` lisab programmile käsu, mille väljund on 1 parajasti variandi 1 korral; väljund salvestatakse mälupeessa 6.
2. `add_and([2, 3])` lisab programmile käsu, mille väljund on 1 parajasti variandi 2 korral; väljund salvestatakse mälupeessa 7.
3. `add_or([6, 7])` lisab programmile käsu, mille väljund on 1 parajasti siis, kui kehtib üks kahest eeltoodud variandist.

Piirangud

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

Alamülesanded

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

6. (8 punkti) Igas kujutises on vasak ülemine piksel (reas 0 veerus 0) must.
7. (14 punkti) $K = 1$.
8. (19 punkti) Lisapiirangud puuduvad.

Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

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

Iga rida peale esimese ja viimase kirjeldab üht kahe musta piksliga kujutist. Ütleme, et rida $2 + i$ kirjeldab kujutist number i . Kujutisel on üks must piksel reas $r_1[i]$ veerus $c_1[i]$ ja teine reas $r_2[i]$ veerus $c_2[i]$.

Hindaja kutsub kõigepealt välja funktsiooni `construct_network(H, W, K)`. Kui `construct_network` rikub mõnda ülesande tekstis toodud tingimust, väljastab hindaja vastava lõigus Realisatsioon kirjeldatud veateate ja lõpetab töö.

Vastasel korral annab hindaja kaks väljundit.

Esiteks väljastab hindaja roboti programmi väljundid:

- rida $1 + i$ ($0 \leq i$): roboti programmi viimase käsu väljund (1 või 0) kujutise number i töötlemisel

Teiseks tekitab hindaja jooksvasse kausta faili `log.txt`:

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

Rida $1 + i$ sisaldab roboti mälu seisuga kujutise number i töötlemise järel: $m[i][j]$ on mälupesas j olev väärtus. Seejuures on mälupesade arv c alati $H \cdot W$ pluss roboti programmi käskude arv.