



Robotski vid

Potrebujemo program za robotski vid. Vsakič ko robotova kamera posname sliko, se ji shrani kot črno-belo sliko v robotovem pomnilniku. Vsaka slika je $H \times W$ mreža slikovnih pik, z vrsticami oštevilčenimi od 0 do $H - 1$ in stolpci od 0 do $W - 1$. Na vsaki sliko obstajata **natanko dve** črni slikovni piki, ostale pike so bele.

Robot lahko obdela vsako sliko s programom sestavljenim iz preprostih ukazov. Dobiš vrednosti H , W in pozitivno celo število K . Tvoja naloga je napisati proceduro, ki sestavi program za robota, ki za poljubno sliko ugotovi, ali je **razdalja** med dvema črnima pikama natanko K . Razdalja med slikovno piko v vrstici r_1 in stolpcu c_1 ter slikovno piko v vrstici r_2 in stolpcu c_2 je $|r_1 - r_2| + |c_1 - c_2|$. V tej formuli predstavlja $|x|$ absolutno vrednost x , kar je enako x če $x \geq 0$ in $-x$ če $x < 0$.

Zdaj opišimo, kako robot deluje.

Robotov pomnilnik je dovolj veliko polje celic, indeksiranih od 0. Vsaka celica hrani ali 0 ali 1. Ko je v celico enkrat zapisana vrednost, se ta več ne spreminja. Slika se shranjena vrstico po vrstico, v celicah z indeksi od 0 do $H \cdot W - 1$. Prva vrstica je shranjena v celicah od 0 do $W - 1$ in zadnja vrstica je v celicah od $(H - 1) \cdot W$ do $H \cdot W - 1$. Torej, slikovna pika v vrstici i in stolpcu j se nahaja v celici $i \cdot W + j$. Bela slikovna pika je predstavlja z 0 in črna z 1.

Robotov program je zaporedje **ukazov**, ki so oštevilčeni z zaporednimi celimi števili, začenši z 0. Ko se program zažene, se ukazi izvajajo eden za drugim. Vsak ukaz prebere vrednosti ene ali več celic (tem vrednostim pravimo **vhod** ukaza) in vrne eno izmed vrednosti 0 ali 1 (tej vrednosti pravimo **izhod** ukaza). Izhod ukaza i je shranjen v celici $H \cdot W + i$. Vhodi ukaza i so lahko celice, ki hranijo slikovne točke, ali izhodi prejšnjih ukazov (tj. celice od 0 do $H \cdot W + i - 1$).

Obstajajo štirje tipi ukazov:

- NOT: ima natanko en vhod. Izhod je 1, če je vhod 0, sicer je izhod 0,
- AND: ima enega ali več vhodov. Izhod je 1, če in samo če so **vs**i vhodi 1,
- OR: ima enega ali več vhodov. Izhod je 1, če in samo če je **vsaj eden** izmed vhodov 1,
- XOR: ima enega ali več vhodov. Izhod je 1, če in samo če je **liho število** vhodov 1.

Izhod zadnjega ukaza programa naj bo 1, če je razdalja med dvema črnima slikovnimama pikama natanko K , sicer naj bo 0.

Podrobnosti implementacije

Impementiraj naslednjo proceduro:

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

- H, W : dimenzije vsake izmed slik, ki jih zajame robotova kamera,
- K : pozitivno celo število,
- Procedura naj ustvari robotov program. Za poljubno zajeto sliko naj program ugotovi, ali je razdalja med dvema črnima pikama slike natanko K .

Procedura naj ključne eno ali več naslednjih funkcij, ki robotovemu programu na konec dodajo ukaze (sprva je program prazen):

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

- Po vrsti te funkcije dodajo ukaze NOT, AND, OR, oz. XOR,
- N (za `add_not`): indeks celice, iz katere dodani ukaz NOT prebere vhod,
- Ns (za `add_and`, `add_or`, `add_xor`): polje, ki vsebuje indekse celic, ki predstavljajo vhod dodanemu ukazu AND, OR, oz. XOR,
- Vsaka funkcija vrne indeks celice, ki hrani izhod tega ukaza. Zaporedni klici teh funkcij vrnejo zaporedna cela števila, začenši z $H \cdot W$.

Robotov program se lahko sestoji iz največ 10 000 ukazov. Ukazi lahko skupaj preberejo največ 1 000 000 vrednosti. Z drugimi besedami: seštevek dolžin polj Ns vseh klicev `add_and`, `add_or` in `add_xor`, ter število klicev `add_not` ne sme preseči 1 000 000.

Po dodajanju zadnjega ukaza, se procedura `construct_network` zaključi. Robotov program bo za tem ocenjen na nekem številu slik. Tvoja rešitev opravi določen testni primer, če za vsako izmed slik velja, da je izhod zadnjega ukaza enak 1, če in samo če je razdalja med dvema črnima slikovnimi pikama enaka K .

Ocenjevanje tvoje rešitve lahko vrne eno izmed naslednjih angleških sporočil o napakah:

- `Instruction with no inputs`: podano je bilo prazno polje kot vhod ukazu `add_and`, `add_or` ali `add_xor`,
- `Invalid index`: podan je bil neveljaven (lahko da negativen) indeks celice kot vhod ukazu `add_and`, `add_or`, `add_xor` ali `add_not`,
- `Too many instructions`: tvoja procedura je poskusila dodati več kot 10 000 ukazov,
- `Too many inputs`: ukazi so skupno prebrali več kot 1 000 000 vrednosti.

Primer

Predpostavimo $H = 2$, $W = 3$ in $K = 3$. Obstajata zgolj dve možni sliki, kjer je razdalja med črnimi slikovnimi pikami 3.

0	1	2
3	4	5

0	1	2
3	4	5

- 1. možnost: črni piki sta 0 in 5,
- 1. možnost: črni piki sta 2 in 3.

Ena izmed možnih rešitev za robotov program je naslednje zaporedje klicev:

1. `add_and([0, 5])`, kar doda ukaz, ki vrne 1, če in samo če velja prva možnost. Izhod je shranjen v celici 6,
2. `add_and([2, 3])`, kar doda ukaz, ki vrne 1, če in samo če velja druga možnost. Izhod je shranjen v celici 7,
3. `add_or([6, 7])`, kar doda ukaz, ki vrne 1, če in samo če velja vsaj ena izmed zgornjih točk.

Omejitve

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

Podnaloge

1. (10 točk) $\max(H, W) \leq 3$,
2. (11 točk) $\max(H, W) \leq 10$,
3. (11 točk) $\max(H, W) \leq 30$,
4. (15 točk) $\max(H, W) \leq 100$,
5. (12 točk) $\min(H, W) = 1$,
6. (8 točk) Pri vsaki sliki je slikovna pika v vrstici 0 in stolpcu 0 črna,
7. (14 točk) $K = 1$,
8. (19 točk) Ni dodatnih omejitev.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v naslednjem formatu:

- vrstica 1: $H \ W \ K$,
- vrstica $2 + i$ ($i \geq 0$): $r_1[i] \ c_1[i] \ r_2[i] \ c_2[i]$,
- zadnja vrstica: -1 .

Vsaka vrstica, izvzemši prvo in zadnjo, predstavlja sliko z dvema črnima slikovnimi pikama. Sliko, opisano v vrstici $2 + i$, označujemo kot sliko i . Ena črna pika je v vrstici $r_1[i]$ in stolpcu $c_1[i]$ ter druga je v vrstici $r_2[i]$ in stolpcu $c_2[i]$.

Vzorčni ocenjevalnik najprej kliče funkcijo `construct_network(H, W, K)`. Če funkcija `construct_network` krši katero izmed omejitev, predstavljeno v opisu problema, vzorčni ocenjevalnik izpiše eno izmed sporočil o napaki, ki so navedene na koncu Podrobnosti implementacije in zaključni.

Če ne ugotovi nobene napake, vzorčni ocenjevalnik vrne dva izhoda.

Najprej vzorčni ocenjevalnik izpiše izhod robotovega programa v naslednjem formatu:

- vrstica $1 + i$ ($0 \leq i$): izhod zadnjega ukaza robotovega programa za sliko i (1 ali 0).

Nato vzorčni ocenjevalnik zapiše datoteko `log.txt` v trenutni mapi, v naslednjim formatu:

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

Zaporedje v vrstici $1 + i$ opisuje vrednosti, shranjene v robotovih pomnilniških celicah, po tem, ko se izvede robotov program in je za vhod podana slika i . Natančneje $m[i][j]$ je vrednost celice j . Opazka: vrednost c (dolžina zaporedja) je enaka seštevku $H \cdot W$ in števila ukazov robotovega programa.