



# Vision Program

**Tekst ovog zadatka izrazito je suhoparan i ne sadrži šale. Savjetujemo vam da ga ne čitate.**

Vaš je zadatak implementirati računalni vid na specifičnom robotu. Svaki puta kada robotske oči (kamere) poslikaju sliku, ona je u memoriji predstavljena pomoću crnih i bijelih piksela. Svaka je slika predstavljena kao dvodimenzionalno polje dimenzija  $H \times W$  piksela, gdje su retci označeni brojevima od 0 do  $H - 1$ , dok su stupci označeni brojevima od 0 do  $W - 1$ . Također, poznato je da se na slici nalaze **točno** dva crna piksela.

Robot sliku može procesirati pomoću programa koji se sastoji od vrlo rudimentarnih operacija.

Dane su vam vrijednosti  $H$ ,  $W$  i pozitivan cijeli broj  $K$ . Vaš je zadatak napisati proceduru koja pronalazi robotski program koji, za svaku sliku koja odgovara danim parametrima, određuje je li **udaljenost** između dvaju crnih piksela jednaka  $K$ . Udaljenost između dvaju piksela od kojih se prvi nalazi u retku  $r_1$  i stupcu  $c_1$ , a drugi se nalazi u retku  $r_2$  i stupcu  $c_2$  iznosi  $|r_1 - r_2| + |c_1 - c_2|$ . Oznaka  $|x|$  u prethodnoj formuli označava apsolutnu vrijednost broja  $x$ , koja iznosi  $x$  ako je  $x \geq 0$ , odnosno  $-x$  ako je  $x < 0$ .

Slijedi opis rada robota.

Memoriju robota zamišljamo kao dovoljno veliko polje koje je indeksirano od 0. Svaki element tog polja može poprimiti vrijednost 0 ili 1, a tu je vrijednost, nakon postavljanja, nemoguće promijeniti. Slika je u toj memoriji spremljena po retcima, a proteže se od indeksa 0 do indeksa  $H \cdot W - 1$ . Preciznije, prvi je redak spremljen od indeksa 0 do indeksa  $W - 1$ , a zadnji je redak spremljen od indeksa  $(H - 1) \cdot W$  do indeksa  $H \cdot W - 1$ . Preciznije, ako je piksel slike u retku  $i$  i stupcu  $j$  crn, vrijednost elementa polja na indeksu  $i \cdot W + j$  iznosi 1. U protivnom, ta vrijednost iznosi 0.

Robotski program je slijed **instrukcija** koje su redom označene cijelim brojevima počevši od 0. Kada se robotski program pokrene, instrukcije se redom izvršavaju (jedna po jedna). Ulaz u svaku instrukciju jest podskup elemenata u memoriji (te vrijednosti nazivamo **ulaznim vrijednostima**), a instrukcija vraća jednu vrijednost koja iznosi 0 ili 1 (tu vrijednost nazivamo **izlaznom vrijednosti**). Izlazna vrijednost  $i$ -te instrukcije u robotskoj memoriji se sprema na indeksu  $H \cdot W + i$ . Ulazne vrijednosti u  $i$ -tu instrukciju mogu biti samo memorijske vrijednosti na indeksima od 0 do  $H \cdot W + i - 1$ .

Postoje četiri vrste instrukcija:

- NOT: prima točno jednu ulaznu vrijednost. Njena izlazna vrijednost iznosi 1 ako ulazna vrijednost iznosi 0, a inače izlazna vrijednost iznosi 0.
- AND: prima jednu ili više ulaznih vrijednosti. Njena izlazna vrijednost iznosi 1 ako i samo ako **sve** njene ulazne vrijednosti iznose 1.
- OR: prima jednu ili više ulaznih vrijednosti. Njena izlazna vrijednost iznosi 1 ako i samo ako **barem jedna** od njenih ulaznih vrijednosti iznosi 1.
- XOR: prima jednu ili više ulaznih vrijednosti. Njena izlazna vrijednost iznosi 1 ako i samo ako **neparno mnogo** njenih ulaznih vrijednosti iznosi 1.

Izlazna vrijednost posljednje izvršene instrukcije robotskog programa treba iznositi 1 ako je udaljenost između crnih piksela sa slike jednaka  $K$ . U protivnom, izlazna vrijednost posljednje izvršene instrukcije treba iznositi 0.

## Implementacijski detalji

Potrebno je implementirati sljedeću proceduru:

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

- $H, W$ : dimenzije slika
- $K$ : pozitivan cijeli broj
- Procedura treba izraditi robotski program opisan u tekstu zadatka. Odnosno, za svaku sliku odgovarajućih dimenzija, robotski program treba odrediti je li udaljenost između crnih piksela točno  $K$ .

Procedura pritom treba pozvati jednu ili više funkcija:

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

- Ove funkcije na kraj robotskog programa dodaju NOT, AND, OR ili XOR instrukciju (redom kako je navedeno).
- $N$  (za `add_not`): indeks u memoriji s kojeg instrukcija NOT čita svoju ulaznu vrijednost.
- $Ns$  (za `add_and`, `add_or`, `add_xor`): polje koje sadrži indekse u memoriji s kojeg instrukcije AND, OR ili XOR čitaju svoje ulazne vrijednosti.
- Svaka funkcija vraća index u memoriji u koju ta ista instrukcija sprema svoju izlaznu vrijednost. Stoga, uzastopni pozivi ovih funkcija će vratiti uzastopne cijele brojeve počevši od  $H \cdot W$ .

Robotski program smije sadržavati najviše 10 000 instrukcija. Instrukcije smiju

**kumulativno** primati najviše 1 000 000 ulaznih vrijednosti. Odnosno, ukupna duljina svih  $N$ s polja u svim pozivima funkcija `add_and`, `add_or` i `add_xor`, zajedno sa brojem poziva funkcije `add_not` ne smije prelaziti 1 000 000.

Nakon poziva posljednje funkcije, procedura `construct_network` treba završiti s izvođenjem. Potom će se taj dobiveni robotski program pokrenuti na određenom broju slika. Vaše rješenje prolazi neki test podatak ako, za svaku takvu sliku, izlaz posljednje izvršene instrukcije iznosi 1 ako i samo ako je udaljenost između crnih piksela slike jednaka  $K$ .

Evaluacija vašeg programa može rezultirati jednom od sljedećih poruka na Engleskom jeziku.

- `Instruction with no inputs`: prazno polje je dano kao ulazni parametar funkcije `add_and`, `add_or` ili `add_xor`.
- `Invalid index`: Neispravan (možda i negativan) indeks se nalazi kao ulaz u `add_and`, `add_or`, `add_xor` ili `add_not` funkciju.
- `Too many instructions`: Vaša procedura je pokušala izgraditi robotski program koji sadrži više od 10 000 instrukcija.
- `Too many inputs`: Instrukcije su kao ulazne podatke kumulativno primile više od 1 000 000 ulaznih vrijednosti.

## Ogledni test podatak

Neka je  $H = 2$ ,  $W = 3$ ,  $K = 3$ . Postoje samo dvije moguće slike gdje je udaljenost između crnih piksela jednaka 3.

0	1	2
3	4	5

0	1	2
3	4	5

- Slučaj 1: crni pikseli su na pozicijama 0 i 5
- Slučaj 2: crni pikseli su na pozicijama 2 i 3

Jedan od mogućih robotskih programa jest:

1. `add_and([0, 5])`, dodaje instrukciju koja vraća 1 ako i samo ako se radi o slučaju 1. Izlaz se nalazi na indeksu 6.
2. `add_and([2, 3])`, dodaje instrukciju koja vraća 1 ako i samo ako se radi o slučaju 2. Izlaz se nalazi na indeksu 7.
3. `add_or([6, 7])`, dodaje instrukciju koja vraća 1 ako i samo ako se radi o slučaju 1 ili slučaju 2.

## Ograničenja

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

## Podzadaci

1. (10 bodova)  $\max(H, W) \leq 3$
2. (11 bodova)  $\max(H, W) \leq 10$
3. (11 bodova)  $\max(H, W) \leq 30$
4. (15 bodova)  $\max(H, W) \leq 100$
5. (12 bodova)  $\min(H, W) = 1$
6. (8 bodova) Piksela u retku 0 i stupcu 0 je crn u svakoj slici.
7. (14 bodova)  $K = 1$
8. (19 bodova) Nema dodatnih ograničenja.

## Ogledni ocjenjivač

Ogledni ocjenjivač čita ulaz u sljedećem formatu:

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

Svi retci osim prvog i zadnjeg predstavljaju sliku sa dva crna piksela. Sliku u retku  $2 + i$  označavamo brojem  $i$ . Jedan crni piksel te slike nalazi se u retku  $r_1[i]$  i stupcu  $c_1[i]$ , dok se drugi nalazi u retku  $r_2[i]$  i stupcu  $c_2[i]$ .

Ogledni ocjenjivač najprije poziva `construct_network(H, W, K)`. Ako ta procedura narušava neko od ograničenja opisanih u tekstu zadatka, ogledni ocjenjivač ispisuje jednu od poruke navedenih na kraju odlomka o implementacijskim detaljima te završava s izvođenjem.

U protivnom, ogledni ocjenjivač ima dva izlaza.

Ogledni ocjenjivač može ispisati poruku `Invalid user input` ako detektira neko odstupanje od formata ulaznih podataka (npr. u ulazu se pojavljuje redak ili stupac koji ne postoji).

Ako takvo odstupanje nije detektirano, ogledni ocjenjivač ispisuje izlaz robotskog programa u sljedećem formatu:

- redak  $1 + i$  ( $0 \leq i$ ): izlaz posljednje instrukcije robotskog programa za sliku  $i$  (1 ili 0).

Dodatno, ogledni ocjenjivač ispisuje sljedeće podatke u datoteku "log.txt" koju stvara u

trenutnoj mapi.

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

Niz brojeva u  $(1 + i)$ -vom retku datoteke predstavlja vrijednosti spremljene u robotovu memoriju nakon izvršavanja robotskog programa nad slikom  $i$ . Preciznije,  $m[j]$  predstavlja vrijednost memorijske lokacije s indeksom  $j$ . Primijetite da duljina niza (c) odgovara broju  $H \cdot W$  uvećanom za broj instrukcija robotskoga programa.