



Vision Program

Vaš je zadatak implementirati računarski vid na specifičnom robotu. Svaki put kada robotske oči (kamere) naprave snimak, on se u memoriji predstavlja pomoću crnih i bijelih piksela. Svaki snimak je predstavljen kao dvodimenzionalni niz dimenzije $H \times W$ piksela, gdje su redovi označeni brojevima od 0 do $H - 1$, dok su kolone označene brojevima od 0 do $W - 1$. Takođe, poznato je da se na slici nalaze **tačno dva** crna piksela.

Robot snimak može procesirati pomoću programa koji se sastoji od vrlo jednostavnih instrukcija.

Date su vrijednosti H , W i pozitivan cio broj K . Vaš je zadatak napisati funkciju koja pronalazi robotski program koji, za svaki snimak koji odgovara datim parametrima, određuje je li **udaljenost** između dva crna piksela jednaka K . Udaljenost između dva piksela od kojih se prvi nalazi u redu r_1 i koloni c_1 , a drugi se nalazi u redu r_2 i koloni 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 veliki niz koje je indeksiran od 0. Svaki element tog niza može imati vrijednost 0 ili 1, a tu je vrijednost, nakon postavljanja, nemoguće promijeniti. Snimak je u toj memoriji sačuvan po redovima, a proteže se od indeksa 0 do indeksa $H \cdot W - 1$. Preciznije, prvi je red sačuvan od indeksa 0 do indeksa $W - 1$, a posljednji je red sačuvan od indeksa $(H - 1) \cdot W$ do indeksa $H \cdot W - 1$. Ako je piksel snimka u redu i i koloni j crn, vrijednost elementa niza na indeksu $i \cdot W + j$ iznosi 1. U suprotnom, ta vrijednost iznosi 0.

Robotski program je niz **instrukcija** koje su redom označene cijelim brojevima počevši od 0. Kada se robotski program pokrene, instrukcije se izvršavaju redom, jedna po jedna. Ulaz u svaku instrukciju je jedan ili više elemenata iz memorije (te vrijednosti nazivamo **ulaznim vrijednostima**), a instrukcija vraća jednu vrijednost koja iznosi 0 ili 1 (tu vrijednost nazivamo **izlaznom vrijednošću**). Izlazna vrijednost i -te instrukcije u robotskoj memoriji se čuva 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 tačno 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 **bar jedna** od njenih ulaznih vrijednosti iznosi 1.
- XOR: prima jednu ili više ulaznih vrijednosti. Njena izlazna vrijednost iznosi 1 ako i samo ako je broj njenih ulaznih vrijednosti koje su jednake 1 **neparan**.

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

Detalji implementacije

Potrebno je implementirati sljedeću funkciju:

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

- H, W : dimenzije snimaka
- K : pozitivan cio broj
- Funkcija treba da kreira robotski program opisan u tekstu zadatka. Preciznije, za svaki snimak odgovarajućih dimenzija, robotski program treba odrediti da li je udaljenost između crnih piksela tačno K .

Funkcija pritom treba pozvati jednu ili više funkcija iz sljedećeg spiska:

```
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 sa kojeg instrukcija NOT čita svoju ulaznu vrijednost.
- Ns (za `add_and`, `add_or`, `add_xor`): niz koje sadrži indekse u memoriji sa kojeg instrukcije AND, OR ili XOR čitaju svoje ulazne vrijednosti.
- Svaka funkcija vraća indeks u memoriji u koju ta ista instrukcija čuva svoju izlaznu vrijednost. Stoga, uzastopni pozivi ovih funkcija će vratiti uzastopne cijele brojeve počevši od $H \cdot W$.

Robotski program može imati najviše 10 000 instrukcija. Instrukcije smiju **kumulativno** primiti najviše 1 000 000 ulaznih vrijednosti. Odnosno, ukupna dužina svih nizova Ns 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, funkcija `construct_network` završava rad. Potom će se taj dobijeni robotski program pokrenuti na određenom broju snimaka. Vaše rješenje prolazi neki test podatak ako, za svaku takav snimak, izlaz posljednje izvršene instrukcije iznosi 1 ako i samo ako je udaljenost između crnih piksela na snimku jednaka K .

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

- `Instruction with no inputs`: prazan niz je dat 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 funkcija je pokušala kreirati 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.

Primjeri

Neka je $H = 2$, $W = 3$, $K = 3$. Postoje samo dva moguća snimka 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 je:

1. `add_and([0, 5])`, dodaje instrukciju koja vraća 1 ako i samo ako se radi o snimku slučaja 1. Izlaz se nalazi na indeksu 6.
2. `add_and([2, 3])`, dodaje instrukciju koja vraća 1 ako i samo ako se radi o snimku slučaja 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 redu 0 i koloni 0 je crn na svakoj slici.
7. (14 bodova) $K = 1$
8. (19 bodova) Nema dodatnih ograničenja.

Program za ocjenjivanje (grader)

Program za ocjenjivanje (grader) učitava podatke u sljedećem formatu:

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

Svi redovi osim prvog i posljednjeg predstavljaju snimke sa dva crna piksela. Snimak u redu $2 + i$ označavamo brojem i . Jedan crni piksel tog snimka nalazi se u redu $r_1[i]$ i koloni $c_1[i]$, dok se drugi crni piksel nalazi u redu $r_2[i]$ i koloni $c_2[i]$.

Program za ocjenjivanje (grader) prvo poziva `construct_network(H, W, K)`. Ako poziv `construct_network` narušava neko od ograničenja iz teksta zadatka, program za ocjenjivanje (grader) štampa jednu od grešaka opisanih u sekciji Detalji implementacije i završava rad. U suprotnom, program za ocjenjivanje (grader) vraća dva izlaza.

Prvo, program za ocjenjivanje (grader) štampa izlaz robotskog programa u sljedećem formatu:

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

Drugo, program za ocjenjivanje (grader) štampa sljedeće podatke u datoteku "log.txt" koju stvara u trenutnom folderu.

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

Niz brojeva u $(1 + i)$ -tom redu datoteke predstavlja vrijednosti sačuvane u robotskoj memoriji nakon izvršavanja robotskog programa nad snimkom i . Preciznije, $m[i][j]$ predstavlja vrijednost memorijske lokacije sa indeksom j . Primijetite da dužina niza (c) odgovara broju $H \cdot W$ uvećanom za broj instrukcija robotskog programa.

