



Program Penglihatan

Anda sedang mengerjakan implementasi sebuah program penglihatan untuk robot. Setiap kali kamera robot mengambil gambar, gambar tersebut disimpan sebagai gambar hitam putih di memori robot. Setiap gambar adalah matriks pixel $H \times W$, dengan baris yang dinomori 0 hingga $H - 1$ dan kolom yang dinomori 0 hingga $W - 1$. Terdapat **persis dua** pixel hitam di setiap gambar, dan pixel lainnya putih.

Robot dapat memroses setiap gambar dengan program yang berisikan instruksi-instruksi sederhana. Anda diberikan nilai-nilai H , W , dan sebuah bilangan bulat K . Gol Anda adalah menulis prosedur untuk menghasilkan program robot yang, untuk gambar apapun, dapat menentukan apakah **jarak** antara dua pixel hitam adalah tepat K . Di sini, jarak antara suatu pixel di baris r_1 dan kolom c_1 dan suatu pixel di baris r_2 dan kolom c_2 adalah $|r_1 - r_2| + |c_1 - c_2|$. Dalam formula ini $|x|$ menyatakan nilai absolut dari x , yang sama dengan x jika $x \geq 0$ dan sama dengan $-x$ jika $x < 0$.

Sekarang kita jelaskan bagaimana cara robot bekerja.

Memori robot adalah array yang berisikan sel-sel dan berukuran cukup besar, dengan indeks mulai dari 0. Setiap sel dapat menyimpan antara 0 atau 1 dan nilainya, sekali di-set, tidak akan berubah. Gambar disimpan baris per baris di sel-sel dengan indeks 0 hingga $H \cdot W - 1$. Baris pertama disimpan di sel-sel 0 hingga $W - 1$, dan baris terakhir disimpan di sel-sel $(H - 1) \cdot W$ hingga $H \cdot W - 1$. Secara khusus, jika pixel di baris i dan kolom j adalah hitam, nilai sel $i \cdot W + j$ adalah 1, jika tidak nilainya 0.

Program robot adalah rangkaian **instruksi**, yang dinomori dengan bilangan bulat berurut mulai dari 0. Saat program dijalankan, instruksi-instruksi di-eksekusi satu persatu. Setiap instruksi membaca nilai-nilai dari satu atau lebih sel (kita sebut nilai-nilai ini sebagai **masukan** instruksi) dan menghasilkan suatu nilai 0 atau 1 (kita sebut nilai ini sebagai **keluaran** instruksi). Keluaran dari instruksi i disimpan di sel $H \cdot W + i$. Masukan dari instruksi i hanya dapat berasal dari sel-sel yang menyimpan pixel atau keluaran instruksi sebelumnya, maksudnya sel-sel 0 hingga $H \cdot W + i - 1$.

Terdapat empat tipe instruksi:

- NOT: memiliki tepat satu masukan. Keluarannya adalah 1 jika masukannya adalah 0, jika tidak keluarannya adalah 0.
- AND: memiliki satu atau lebih masukan. Keluarannya adalah 1 jika dan hanya jika **semua** masukannya adalah 1.
- OR: memiliki satu atau lebih masukan. Keluarannya adalah 1 jika dan hanya jika **setidaknya satu** masukannya adalah 1.

- XOR: memiliki satu atau lebih masukan. Keluarannya adalah 1 jika dan hanya jika banyaknya 1 pada masukan adalah **ganjil** .

Keluaran dari instruksi terakhir dari program haruslah 1 jika jarak antara dua pixel hitam adalah tepat K , dan 0 jika tidak.

Rincian implementasi

Anda harus melakukan implementasi untuk prosedur berikut:

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

- H, W : dimensi dari setiap gambar yang diambil oleh kamera robot
- K : suatu bilangan bulat
- Prosedur ini harus menghasilkan program robot. Untuk setiap gambar yang diambil oleh kamera robot, program ini harus menentukan apakah jarak antara dua pixel hitam pada gambar adalah tepat K .

Prosedur ini dapat memanggil satu atau lebih dari prosedur-prosedur berikut untuk menambahkan instruksi ke program robot (yang awalnya kosong):

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

- Menambahkan suatu instruksi NOT, AND, OR, atau XOR, secara berturutan.
- N (untuk `add_not`): indeks sel dari mana instruksi NOT yang ditambahkan mendapatkan masukannya
- Ns (untuk `add_and`, `add_or`, `add_xor`): array berisikan indeks sel-sel dari mana instruksi AND, OR, or XOR yang ditambahkan mendapatkan masukannya
- Setiap prosedur mengembalikan indeks sel yang menyimpan keluaran dari instruksi. Setiap panggilan prosedur yang berturutan mengembalikan bilangan bulat yang berturutan dimulai dari $H \cdot W$.

Program robot dapat berisikan hingga 10 000 instruksi. Instruksi-instruksi dapat membaca hingga sebanyak total 1 000 000 nilai. Dengan kata lain, total panjang dari array-array Ns di semua panggilan `add_and`, `add_or` dan `add_xor` ditambah dengan banyaknya panggilan `add_not` tidak dapat melebihi 1 000 000.

Setelah menambahkan instruksi terakhir, prosedur `construct_network` harus berakhir (`return`). Program robot kemudian akan dievaluasi menggunakan beberapa gambar. Solusi Anda berhasil melalui suatu kasus uji jika untuk setiap gambar tersebut, keluaran dari instruksi terakhir adalah 1 jika dan hanya jika jarak antara dua pixel hitam pada gambar adalah K .

Pengujian solusi Anda mungkin menghasilkan salah satu pesan error berikut:

- `Instruction with no inputs`: suatu array kosong telah diberikan sebagai masukan terhadap `add_and`, `add_or`, atau `add_xor`.
- `Invalid index`: suatu indeks sel yang salah (mungkin negatif) telah diberikan sebagai masukan terhadap `add_and`, `add_or`, `add_xor`, atau `add_not`.
- `Too many instructions`: prosedur Anda mencoba menambahkan lebih dari 10 000 instruksi.
- `Too many inputs`: instruksi mencoba membaca lebih dari 1 000 000 nilai secara total.

Contoh

Misalkan $H = 2$, $W = 3$, $K = 3$. Hanya ada dua kemungkinan gambar dimana jarak antara pixel hitam adalah 3.

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | 4 | 5 |

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | 4 | 5 |

- Kasus 1: pixel hitam adalah 0 dan 5
- Kasus 2: pixel hitam adalah 2 dan 3

Salah satu solusi adalah membuat program robot dengan panggilan-panggilan berikut:

1. `add_and([0, 5])`, yang menambahkan instruksi yang mengeluarkan 1 jika dan hanya jika kasus pertama berlaku. Keluarannya disimpan di sel 6.
2. `add_and([2, 3])`, yang menambahkan instruksi yang mengeluarkan 1 jika dan hanya jika kasus kedua berlaku. Keluarannya disimpan di sel 7.
3. `add_or([6, 7])`, yang menambahkan instruksi yang mengeluarkan 1 jika dan hanya jika salah satu kasus berlaku.

Batasan

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

Subsoal

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

3. (11 poin) $\max(H, W) \leq 30$
4. (15 poin) $\max(H, W) \leq 100$
5. (12 poin) $\min(H, W) = 1$
6. (8 poin) Pixel di baris 0 dan kolom 0 adalah hitam di setiap gambar.
7. (14 poin) $K = 1$
8. (19 poin) Tidak ada batasan tambahan.

Grader contoh

Grader contoh membaca masukan dengan format sebagai berikut:

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

Setiap baris kecuali baris pertama dan terakhir menyatakan suatu gambar dengan dua pixel hitam. Kita sebut gambar yang dinyatakan di baris $2 + i$ sebagai gambar i . Satu pixel hitam ada di baris $r_1[i]$ dan kolom $c_1[i]$ dan satunya lagi di baris $r_2[i]$ dan kolom $c_2[i]$.

Grader contoh pertama memanggil `construct_network(H, W, K)`. Jika `construct_network` melanggar batasan yang dijelaskan di pernyataan soal, grader contoh akan mencetak salah satu pesan error yang diberikan di bagian akhir bagian Rincian implementasi dan berakhir.

Jika tidak, grader contoh menghasilkan dua keluaran.

Pertama, grader contoh mencetak keluaran program robot dengan format sebagai berikut:

- baris $1 + i$ ($0 \leq i$): keluaran dari instruksi terakhir dari program robot untuk gambar i (1 atau 0).

Kedua, grader contoh membuat sebuah berkas `log.txt` di direktori yang sama dengan format sebagai berikut:

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

Rangkaian pada baris $1 + i$ menjelaskan nilai-nilai yang disimpan di sel-sel memori robot setelah program robot dijalankan, dengan gambar i sebagai masukan. Secara khusus, $m[i][j]$ merupakan nilai sel j . Perhatikan bahwa nilai c (panjang rangkaian) sama dengan $H \cdot W$ ditambah dengan banyaknya instruksi pada program robot.