



Görüntü İşleme Programı

Bir robot için bir görüntü işleme programı kodlayacaksınız. Robotun kamerası bir fotoğraf çektiğinde, bu fotoğraf robotun hafızasında siyah ve beyaz piksellerden oluşan bir görüntü olarak saklanmaktadır. Her bir görüntü $H \times W$ boyunda, piksellerden oluşan bir ızgaradır ve satırlar 0'dan $H - 1$ 'e, sütunlar da 0'dan $W - 1$ 'e numaralandırılmıştır. Her bir görüntüde **tam olarak iki tane** siyah piksel vardır ve diğer bütün pikseller beyazdır.

Robot, her bir görüntüyü basit komutlardan oluşan bir program ile işleyebilir. Size H , W ve pozitif bir tamsayı olan K değerleri verilecektir. Hedefiniz robot için bir görüntü işleme programı oluşturacak bir prosedür yazmaktır; öyle ki, herhangi bir görüntü için, bu robot programı, iki siyah piksel arasındaki **mesafenin** tam olarak K 'ye eşit olup olmadığını belirleyecektir. r_1 satırında ve c_1 sütununda olan bir piksel ile r_2 satırında ve c_2 sütununda olan diğer bir piksel arasındaki mesafe $|r_1 - r_2| + |c_1 - c_2|$ 'dir. Bu formülde $|x|$, x 'in mutlak değerini belirtmektedir ve $x \geq 0$ ise x 'e, $x < 0$ ise de $-x$ 'e eşittir.

Şimdi robotun nasıl çalıştığını açıklayacağız.

Robotun hafızası yeterli büyüklükte bir hücreler dizisidir ve hücreler 0'dan başlayarak indekslenmiştir. Her bir hücreye ya 0 ya da 1 değeri konulabilir ve bir hücreye bir değer konulduktan sonra o değer değiştirilmeyecektir. Görüntü, hafızada satır satır 0 indeksli hücreden $H \cdot W - 1$ indeksli hücreye kaydedilmiştir. Görüntünün ilk satırı 0 hücrelerinden $W - 1$ hücrelerine, son satırı da $(H - 1) \cdot W$ hücrelerinden $H \cdot W - 1$ hücrelerine kaydedilmiştir. Diğer bir deyişle, eğer i satır ve j sütundaki bir piksel siyah ise $i \cdot W + j$ hücrelerinin değeri 1'dir, beyazsa da 0'dır.

Bir robot programı bir dizi **komuttan** oluşmaktadır ve 0'dan başlayarak ardışık tamsayılarla numaralandırılmışlardır. Program çalıştırıldığında komutlar birer birer çalıştırılırlar. Her bir komut bir ya da daha fazla hücredeki değerleri okur (bu değerlere o komutun **girdileri** diyoruz) ve sonra 0 ya da 1'e eşit olan tek bir değer üretir (bu değere o komutun **çıkması** diyoruz). i . komutun çıkması $H \cdot W + i$ hücrelerine kaydedilir. i . komutun girdileri ya piksel değerlerinin bulunduğu hücreler ya da daha önce çalıştırılan komutların çıktıları olabilirler, yani i . komutun girdileri 0'dan $H \cdot W + i - 1$ hücrelerine kadar olan hücreler olabilir (sondaki hücre dahil).

Dört tip komut bulunmaktadır:

- NOT: sadece bir girdisi vardır. Eğer girdi 0 ise çıkması 1'dir, değil ise çıktı 0'dır.

- **AND**: bir ya da daha fazla girdisi vardır. Yalnızca eğer girdilerin **hepsi** 1 ise çıktısı 1'dir.
- **OR**: bir ya da daha fazla girdisi vardır. Yalnızca eğer girdilerin **en az bir tanesi** 1 ise çıktısı 1'dir.
- **XOR**: bir ya da daha fazla girdisi vardır. Yalnızca eğer girdilerdeki 1'lerin sayısı bir **tek sayı ise** çıktısı 1'dir.

Programdaki son komutun çıktısı, eğer iki siyah piksel arasındaki mesafe tam olarak K ise 1 değil ise 0 olmalıdır.

Kodlama detayları

Aşağıdaki prosedürü kodlamalısınız:

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

- H, W : robotun kamerası ile çekilen her bir görüntünün boyutları
- K : pozitif bir tam sayı
- Bu prosedür bir robot programı üretmelidir. Robotun kamerası ile çekilen herhangi bir görüntü için bu program, görüntüdeki iki siyah piksel arasındaki mesafenin tam olarak K 'ye eşit olup olmadığını belirlemelidir.

Yazacağınız prosedür, aşağıda verilen prosedürleri bir ya da daha fazla çağırarak ilk başta boş olan robot programına komutlar eklemelidir.:

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

- Bu prosedürler NOT, AND, OR, ya da XOR komutlarını eklerler.
- N (add_not için): eklenen NOT komutunun girdisini okuyacağı hücrenin indeksidir
- Ns (add_and, add_or, add_xor için): eklenen AND, OR, ya da XOR komutunun girdilerinin okunacağı hücrelerin indekslerini içeren bir dizidir
- Her prosedür, ilgili komutun çıktısının yazıldığı hücrenin indeksini döndürür. Bu prosedürlere yapılan ardışık çağrılar $H \cdot W$ 'den başlayarak ardışık tamsayılar döndürürler.

Robot programı en fazla 10 000 komuttan oluşabilir. Bu komutlar en fazla toplam 1 000 000 değer okuyabilirler. Diğer bir deyişle, add_and, add_or ve add_xor çağrılarındaki Ns dizilerinin toplam uzunluğu artı add_not çağrılarının toplamı 1 000 000'u geçemez.

Son komutu da ekledikten sonra construct_network prosedürü sonlanmalıdır. Daha sonra robot programı bazı görüntüler üzerinde değerlendirilecektir. Verilen bir test'te,

o test'teki her bir resim için, resimdeki siyah pikseller arasındaki mesafe sadece K 'ya eşit olduğunda son komutun çıktısı 1 ise çözümünüz o test'i geçer.

Çözümünüzün değerlendirilmesi aşağıdaki hata mesajlarından birisi ile sonuçlanabilir:

- Instruction with no inputs: `add_and`, `add_or`, ya da `add_xor` komutlarına boş bir dizi girdi olarak verilmiştir.
- Invalid index: yanlış (belki negatif) bir hücre indeksi `add_and`, `add_or`, `add_xor`, ya da `add_not` komutuna girdi olarak verilmiştir.
- Too many instructions: prosedürünüz 10 000'den fazla komut eklemeye çalışmıştır.
- Too many inputs: komutlar toplamda 1 000 000'den fazla değer okumuşturlar.

Örnek

$H = 2$, $W = 3$, ve $K = 3$ olsun. İki siyah piksel arasındaki mesafenin 3 olduğu yalnızca iki tane olası görüntü vardır.

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

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

- Durum 1: siyah pikseller 0 ve 5'tir
- Durum 2: siyah pikseller 2 ve 3'tür

Olası bir çözüm robotun programını aşağıdaki çağrılarını yaparak oluşturur:

1. `add_and([0, 5])`, bu çağrı ancak Durum 1 geçerli olduğunda 1 çıktısı verecek bir komut ekler. Çıktı 6 numaralı hücreye yazılır.
2. `add_and([2, 3])`, bu çağrı ancak Durum 2 geçerli olduğunda 1 çıktısı verecek bir komut ekler. Çıktı 7 numaralı hücreye yazılır.
3. `add_or([6, 7])`, bu çağrı yukarıdaki durumlardan birisi gerçekleştiğinde 1 çıktısı veren bir komut ekler.

Kısıtlar

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

Alt görevler

1. (10 puan) $\max(H, W) \leq 3$

2. (11 puan) $\max(H, W) \leq 10$
3. (11 puan) $\max(H, W) \leq 30$
4. (15 puan) $\max(H, W) \leq 100$
5. (12 puan) $\min(H, W) = 1$
6. (8 puan) Her bir resimdeki 0 satır ve 0 sütundaki piksel siyahtır.
7. (14 puan) $K = 1$
8. (19 puan) Ek kısıt bulunmamaktadır.

Örnek puanlayıcı

Örnek puanlayıcı girdiyi aşağıdaki formatta okur:

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

İlk ve son satır dışındaki diğer bütün satırlar iki tane siyah pixel buluan bir görüntüyü belirtirler. $2 + i$. satırda belirtilen görüntüye, görüntü i diyelim. Siyah piksellerden birisi $r_1[i]$ satır ve $c_1[i]$ sütunundadır ve diğeri de $r_2[i]$ satır ve $c_2[i]$ sütunundadır.

Örnek puanlayıcı ilk olarak `construct_network(H, W, K)` prosedürünü çağırır. Eğer `construct_network` çağrısı problem tanımındaki kısıtlara uymazsa, örnek puanlayıcı kodlama detayları sonunda listelenen hata mesajlarından birini basıp çıkar.

Aksi takdirde örnek puanlayıcı iki çıktı üretir.

İlk olarak, örnek puanlayıcı robot programın çıktısını aşağıdaki formatta basar:

- satır $1 + i$ ($0 \leq i$): i görüntüsü için robot programın son komutunun çıktısı (1 ya da 0'dır).

İkinci olarak, örnek puanlayıcı o anki dizinde `log.txt` adlı bir dosyayı aşağıdaki formatta oluşturur:

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

Satır $1 + i$ 'deki dizi robot programı i görüntüsü girdi olarak verilip çalıştırıldıktan sonra robot hafızasındaki hücrelerin değerlerini gösterir. Spesifik olarak, $m[i][j]$, j hücresinin değerini gösterir. c yani satırdaki dizinin uzunluğunun $H \cdot W$ artı robot programındaki komutların sayısına eşit olduğuna dikkat ediniz.