



# Mekanik Oyuncak Bebek

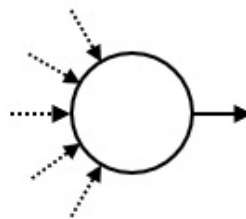
Bir mekanik oyuncak bebek (veya kısaca mekanik bebek), belirli bir dizi (sequence) hareketi, otomatik olarak tekrarlayan bir bebektir. Japonya'da, tarih öncesi zamanlardan beri çok sayıda mekanik bebek yapılmıştır.

Bir mekanik bebeğin hareketleri, **cihazlardan** (device) oluşan bir **devre** (circuit) tarafından yönetilir. Cihazlar tüplerle birbirine bağlıdır. Her cihazın bir veya iki tane **çıkışı** vardır ve rastgele sayıda (sıfır da olabilir) **girişi** olabilir. Her cihazın, herhangi sayıda girişi olabilir. Her tüp, bir cihazın çıkışını, aynı veya başka bir cihazın girişine bağlar. Her girişe tam olarak bir tüp ve her çıkışa tam olarak bir tüp bağlıdır.

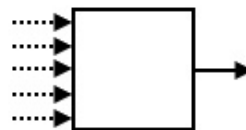
Bebeğin nasıl hareket ettiğini anlamak için, cihazların birine yerleştirilmiş bir **top** düşünün. Top devre boyunca yolculuk etmektedir. Bu yolculuğun her adımında, top cihazı bir çıkışından terk eder, çıkışa bağlı olan tüpü takip eder ve tüpün diğer ucuna bağlı olan cihaza girer.

Üç çeşit cihaz bulunmaktadır: **başlangıç** (origin), **tetik** (trigger) ve **anahtar** (switch). Tam olarak bir adet başlangıç,  $M$  adet tetik ve  $S$  adet anahtar bulunmaktadır ( $S$  sıfır olabilir).  $S$ 'nin değerine karar vermelisiniz. Her cihazın eşsiz (unique) bir seri numarası vardır.

Başlangıç, topun ilk olarak konumlandırıldığı cihazdır. Bir çıkışı vardır. Seri numarası 0'dır.

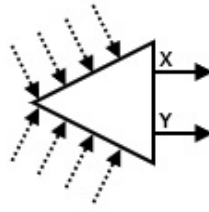


Bir tetik, top kendisine girdiğinde, bebeğin belirli bir hareket yapmasına yol açar. Her tetiğin bir çıkışı vardır. Tetiklerin seri numaraları 1'den  $M$ 'e kadardır.



Her anahtarın, 'X' and 'Y' isminde iki çıkışı vardır. Anahtarın **durumu** (state) 'X' veya 'Y' dir. Top bir anahtara girdikten sonra, anahtarın durumu ile aynı olan çıkıştan

anahtarı terk eder. Bundan sonra, anahtar, durumunu zıt duruma deęiřtirir. En bařta, bütn anahtarların durumu 'X' dir. Anahtarların seri numaraları  $-1$ 'den  $-S$ 'e kadardır.



Size, tetiklerin sayısı,  $M$  veriliyor. Her elemanı bir tetikn seri numarasına karřılık gelen,  $N$  uzunluęunda bir de  $A$  dizisi (sequence) veriliyor. Her tetik,  $A$  dizisinde birkaç kez (sıfır da olabilir) geebilir. Greviniz, ařaęıdaki kořulları saęlayan bir devre tasarlamaktır:

- Top, belli sayıda adımdan sonra bařlangı noktasına dner.
- Top bařlangıca ilk dndęnde, btn anahtarların durumu 'X' olur.
- Top, tam olarak  $N$  defa tetiklere girdikten sonra ilk olarak bařlangıca dner. Bu tetiklerin seri numaraları, topun tetiklere girme sırasına gre,  $A_0, A_1, \dots, A_{N-1}$  'dir.
- Top ilk kez bařlangıca dnmeden nce, btn anahtarların top tarafından yapılan durum deęiřikliklerinin toplam sayısına  $P$  diyelim.  $P$  'nin deęeri 20 000 000'u ařamaz.

Aynı zamanda, ok fazla anahtar kullanmak istemiyorsunuz.

## Kodlama detayları

Ařaęıdaki prosedr kodlamalısınız.

```
create_circuit(int M, int[] A)
```

- $M$ : tetiklerin sayısı.
- $A$ : topun hangi sırada tetiklere girmesi gerektięini gsteren  $N$  uzunluęunda bir tetik seri numaraları dizisi.
- Prosedr tam olarak bir kez aęrılır.
- $N$  'nin  $A$  dizisinin uzunluęu olduęuna ve Programlama Duyuruları dokmanında belirtildięi gibi elde edilebileceęine dikkat ediniz.

Programınız cevap vermek iin ařaęıdaki prosedr aęırmalıdır.

```
answer(int[] C, int[] X, int[] Y)
```

- $C$ :  $M + 1$  uzunluęunda bir dizi.  $i$  ( $0 \leq i \leq M$ ) cihazının ıkıřı  $C[i]$  cihazına baęlıdır.
- $X, Y$ : aynı boyda diziler. Bu dizilerin uzunluęu olan  $S$  anahtarların sayısını

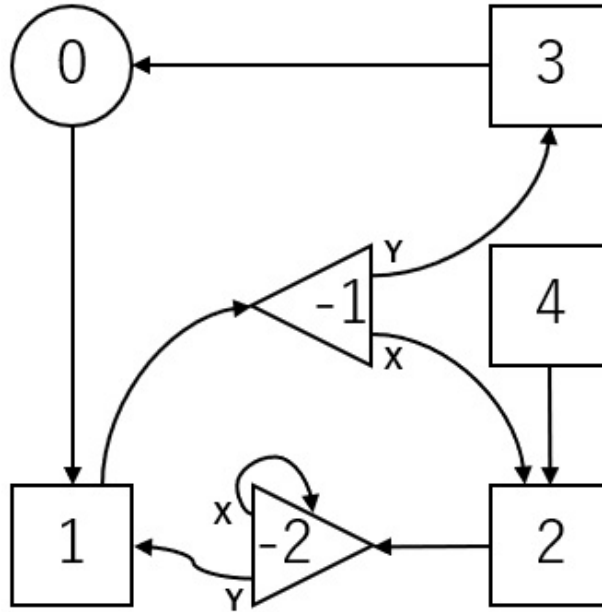
göstermektedir.  $-j$  ( $1 \leq j \leq S$ ) anahtarı için, anahtarın 'X' çıkışı  $X[j - 1]$  cihazına bağlıdır ve 'Y' çıkışı  $Y[j - 1]$  cihazına bağlıdır.

- C, X, ve Y'nin her bir elemanı  $-S$  ve  $M$  arasında (sınırlar dahil) bir tamsayı olmalıdır.
- $S$  en fazla 400 000 olabilir.
- Bu prosedür tam olarak bir kez çağrılmalıdır.
- C, X, ve Y tarafından belirtilen devre problem tanımında belirtilen bütün kısıtları sağlamalıdır.

Yukarıdaki bazı kısıtlar sağlanmazsa programınız **Wrong Answer** (Yanlış Cevap) olarak değerlendirilir. Kısıtların hepsi sağlanırsa, programınız **Accepted** (Kabul Edildi) olarak değerlendirilir ve puanınız  $S$ 'ye göre hesaplanır (Altgörevlere bakınız).

## Örnek

$M = 4$ ,  $N = 4$ , ve  $A = [1, 2, 1, 3]$  olsun. Değerlendirici `create_circuit(4, [1, 2, 1, 3])` prosedürünü çağırır.



Yukarıdaki şekil `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])` prosedür çağırışı ile tanımlanmış bir devre göstermektedir. Şekildeki sayılar cihazların seri numaralarıdır.

İki anahtar kullanılmıştır. O nedenle,  $S = 2$ 'dir.

Başlangıçta,  $-1$  ve  $-2$  anahtarlarının her ikisinin de durumu 'X'tir.

Top aşağıdaki gibi hareket eder:

$$0 \longrightarrow 1 \xrightarrow{X} 2 \xrightarrow{X} -2 \xrightarrow{Y} 1 \xrightarrow{Y} 3 \longrightarrow 0$$

- Top  $-1$  anahtarına ilk kez girdiğinde durumu 'X'tir. Bu nedenle, top 2 tetiğine doğru hareket eder. Daha sonra  $-1$  anahtarının durumu 'Y' olur.

- Top  $-1$  anahtarına ikinci kez girdiğinde durumu 'Y'dir. Bu nedenle, 3 tetiğine doğru hareket eder. Daha sonra  $-1$  anahtarının durumu 'X' olur.

Top ilk kez başlangıca 1,2,1,3 tetiklerine girmiş olarak geri döner.  $-1$  ve  $-2$  anahtarlarının her ikisinin de durumu 'X'tir.  $P$ 'nin değeri 4 olur. Bu nedenle, bu devre kısıtların hepsini sağlar.

Ekteki ziplenmiş paketteki `sample-01-in.txt` dosyası bu örneğe karşılık gelmektedir. Pakette bundan başka örnek girdiler de bulunmaktadır.

## Kısıtlar

- $1 \leq M \leq 100\,000$
- $1 \leq N \leq 200\,000$
- $1 \leq A_k \leq M$  ( $0 \leq k \leq N - 1$ )

## Altgörevler

Her bir test case'in puanı ve kısıtları aşağıdaki gibidir:

1. (2 points) Her bir  $i$  ( $1 \leq i \leq M$ ) için,  $i$  tamsayısı  $A_0, A_1, \dots, A_{N-1}$  dizisinde en fazla bir kez geçer.
2. (4 points) Her bir  $i$  ( $1 \leq i \leq M$ ) için,  $i$  tamsayısı  $A_0, A_1, \dots, A_{N-1}$  dizisinde en fazla iki kez geçer.
3. (10 points) Her bir  $i$  ( $1 \leq i \leq M$ ) için,  $i$  tamsayısı  $A_0, A_1, \dots, A_{N-1}$  dizisinde en fazla 4 kez geçer.
4. (10 points)  $N = 16$
5. (18 points)  $M = 1$
6. (56 points) Ek kısıt bulunmamaktadır

Her bir test case için, programınız **Accepted** (Kabu Edildi) olarak değerlendirilmişse, puanınız  $S$ 'nin değerine göre aşağıdaki gibi hesaplanır:

- Eğer  $S \leq N + \log_2 N$  ise, bu test case için tam puan alırsınız.
- Altgörev 5 ve 6'daki her bir test case için, eğer  $N + \log_2 N < S \leq 2N$  ise kısmi bir puan alırsınız. Bu test case için puanınız,  $0.5 + 0.4 \times \left( \frac{2N - S}{N - \log_2 N} \right)^2$  değerinin o altgörev için belirlenmiş puan ile çarpılmasıyla elde edilir.
- $S$ 'nin değeri yukardaki durumların dışında ise puanınız 0 olur.

Her bir altgörev için puanınız o altgörevdeki bütün test case'lerden alınan puanların minimumuna eşittir.

## Örnek değerlendirici

Örnek değerlendirici girdiyi standart girdiden aşağıdaki formatta okur.

- satır 1:  $M N$
- satır 2:  $A_0 A_1 \dots A_{N-1}$

Örnek değerlendirici üç çıktı üretir.

İlk olarak, örnek değerlendirici cevabınızı `out.txt` isimli bir dosyaya aşağıdaki formatta yazar:

- satır 1:  $S$
- satır  $2 + i$  ( $0 \leq i \leq M$ ):  $C[i]$
- satır  $2 + M + j$  ( $1 \leq j \leq S$ ):  $X[j - 1] Y[j - 1]$

İkinci olarak, örnek değerlendirici topun hareketlerini simüle eder. Sırasıyla topun girdiği cihazların seri numaralarını `log.txt` isimli dosyaya yazar.

Üçüncü olarak, örnek değerlendirici cevabınızın değerlendirme sonucunu standart çıktıya yazar.

- Programınız **Accepted** (Kabul Edildi) olarak değerlendirilmişse örnek değerlendirici  $S$  ve  $P$  değerlerini `Accepted: S P` olarak basar.
- Programınız **Wrong Answer** (Yanlış Cevap) olarak değerlendirilmişse örnek değerlendirici `Wrong Answer: MSG` basar. MSG'nin anlamı aşağıdaki gibidir:
  - `answered not exactly once`: answer prosedürü tam olarak bir kez çağrılmamıştır.
  - `wrong array length`: C'nin uzunluğu  $M + 1$  değildir, ya da X ve Y dizilerinin uzunlukları birbirlerinden farklıdır.
  - `over 400000 switches`:  $S$ , 400 000'den büyüktür.
  - `wrong serial number`: C, X, ya da Y dizilerindeki elemanlardan birisi ya  $-S$ 'den küçüktür ya da  $M$ 'den büyüktür.
  - `over 20000000 inversions`: Top 20 000 000 durum değişikliği içinde başlangıca dönmemiştir.
  - `state 'Y'`: Top ilk kez başlangıca döndüğünde durumu 'Y' olan bir cihaz vardır.
  - `wrong motion`: Hareketlere neden olaran tetikler  $A$  dizisinden farklıdır.

Programınız `Wrong Answer` (Yanlış Cevap) olarak değerlendirildiğinde `out.txt` ve/veya `log.txt` dosyalarının üretilmeyebileceğine dikkat ediniz.