

# Hararetli Toplantı

**Zaman kısıtı:** 2 sn

**Memory kısıtı:** 256 MB

5. IOI altın madalyasını kazanmak için kamplarda hangi konuların anlatılması gerektiğini tartışmak için toplantı masasına oturan rehberler, en sonunda hararetli bir biçimde birbirlerini sırayla masadan kovmaya başladılar.

Masada oturan  $N$  adet rehber olduğunu ve  $i$  numaralı rehberin gücünün  $C_i$  olduğunu biliyorsunuz. Bu rehberlerin her biri başlangıçta kendi numarası yazılı bir koltukta oturmaktadır ve bazı koltuklar arasında doğrudan yollar bulunmaktadır. Bu koltuk ve yollar bir çizge olarak düşünülürse **her rehber ikilisinin yolları kullanarak birbirlerine ulaşabildikleri, bu çizgedeki hiçbir düğümün birden fazla basit döngüde bulunmayacağı ve herhangi bir koltuktan kendine direkt yol olmadığı garanti edilmiştir. Ancak çizgede iki koltuk arasında birden fazla yol bulunabilir.**

Masada tek bir rehber kalana kadar aşağıdaki işlem uygulanarak her seferinde bir rehber masadan kovulmaktadır:

Öncelikle, bir rehber (Rehber  $a$ ) söz hakkı alarak bulunduğu koltuğa bağlı yollardan birini seçerek bu yolun diğer tarafında bulunan rehber (Rehber  $b$ ) suçlamada bulunur. Bu işlem tartışmanın şiddetini  $C_a$  kadar artırır. Rehber  $b$  buna çok üzülerek koltuğunu terk eder ve Rehber  $a$  onun koltuğuna oturur. Rehber  $a$ 'nın eski koltuğundaki yolların bu yeni koltuktan erişilemediğini unutmayın. İşlem sonunda  $a$  numaralı rehberin gücü  $C_b$  kadar artar.

Bu işlemler esnasında dikkat edilmesi gereken, masadaki tartışmanın şiddetinin asla azalmamasıdır. Yani ilk işlem haricindeki her işlem en az bir önceki işlem kadar tartışma gücünü artırmalıdır. Bir önceki işlemle artış miktarı eşit ise de suçlama yapan rehberin numarasının en az bir önceki işlemde suçlama yapan rehber kadar olması gerekmektedir. Diğer bir deyişle,  $k > 1$  numaralı suçlamada  $C_{a_k} \geq C_{a_{k-1}}$  olmalıdır.

$C_{a_k} = C_{a_{k-1}}$  olması durumunda ise  $a_k \geq a_{k-1}$  sağlanmak zorundadır. Burada bahsedilen

$a_i$  değeri,  $i$ .suçlamayı yapan rehberin numarasıdır.

Sizden istenen, bu işlemlerin şartlara uygun olarak yapıldığını ve masada bir rehber kaldığını bildiğinize göre işlemlerin kaç farklı şekilde yapılabileceği ve tüm ihtimaller için tartışma güçlerinin toplamını hesaplamanızdır. Bu değerler çok büyük olabileceği için cevabın **998244353** sayısına bölümünden kalanı hesaplamanız gerekmektedir. Eğer tartışmanın herhangi bir anında söz hakkı alan rehber veya seçtiği yol farklıysa iki tartışma birbirinden farklı sayılmaktadır. Unutmayınız ki bir rehber birden fazla yolla başka bir rehberi suçlayabiliyorsa her biri farklı birer ihtimal sayılmaktadır.

## İmplementasyon Detayı:

Sizden istenen, aşağıdaki fonksiyonu implement etmenizdir. Program bir kere çalıştırıldığında fonksiyon birden fazla kere çağrılabilir.

```
toplanti (N, M, C[], U[], V[])
```

Bu fonksiyon parametre olarak sırasıyla rehber sayısını, koltuklar arasındaki yolların sayısını, rehberlerin gücünü belirten  $N$  elemanlı diziye ve koltuklar arasındaki yolları belirten  $M$  elemanlı  $U$  ve  $V$  dizilerini almaktadır.  $i$  numaralı yol  $U_i$  ile  $V_i$  numaralı koltuklar arasında yer almaktadır. Verilen kenarların yukarıda belirtilmiş olan şartları sağladığı garanti edilmektedir. Cevabınızı hesapladıktan sonra iki elemanlı bir dizi döndürmeniz beklenmektedir. Bu dizinin ilk elemanı kaç farklı şekilde tartışmaların sonlanabileceğini, ikinci elemanı ise olası tüm tartışmalar için tartışma şiddetlerinin toplamını içermelidir. Daha fazla bilgi için verilmiş olan `toplanti-attachments.zip` içindeki dosyaları kontrol edebilirsiniz.

## Grader Kullanımı İçin Girdi ve Çıktı:

Girdinin ilk satırı `toplanti` fonksiyonunun kaç kere çağrılacağını, diğer bir deyişle test senaryolarının sayısını içerir.

Her test senaryosu için girdi formatı aşağıdaki gibidir:

```
N M
C[1] C[2] ... C[N]
U[1] V[1]
U[2] V[2]
...
U[M] V[M]
```

Her test senaryosu için çıktı boşlukla ayrılmış iki tamsayı içermektedir. Sırasıyla farklı ihtimallerin sayısı ve her ihtimal için tartışma şiddetleri toplamı.

## Kısıtlar

- $\Sigma N, \Sigma M$  her fonksiyon çağırısı için değerlerin toplamı olsun.
- $1 \leq N, \Sigma N \leq 2 * 10^5$
- $0 \leq M, \Sigma M \leq 2 * 10^5$
- $0 \leq C_i \leq 10^9$
- $1 \leq U_i, V_i \leq N$

## Alt Görevler

### Alt Görev 1 [7 puan]

$$N \leq 7, \Sigma N \leq 10^4$$

### Alt Görev 2 [2 puan]

$$M = N - 1$$

$$C_i = 0$$

### Alt Görev 3 [4 puan]

$$C_i = 0$$

### Alt Görev 4 [11 puan]

$$2 \leq N, M = N$$

$$U_i = i, V_i = (i \bmod N) + 1$$

### Alt Görev 5 [14 puan]

$$M = N - 1$$

### Alt Görev 6 [17 puan]

$$N, \Sigma N \leq 3000$$

### Alt Görev 7 [45 puan]

Ek kısıt yoktur.

## Örnek Girdi

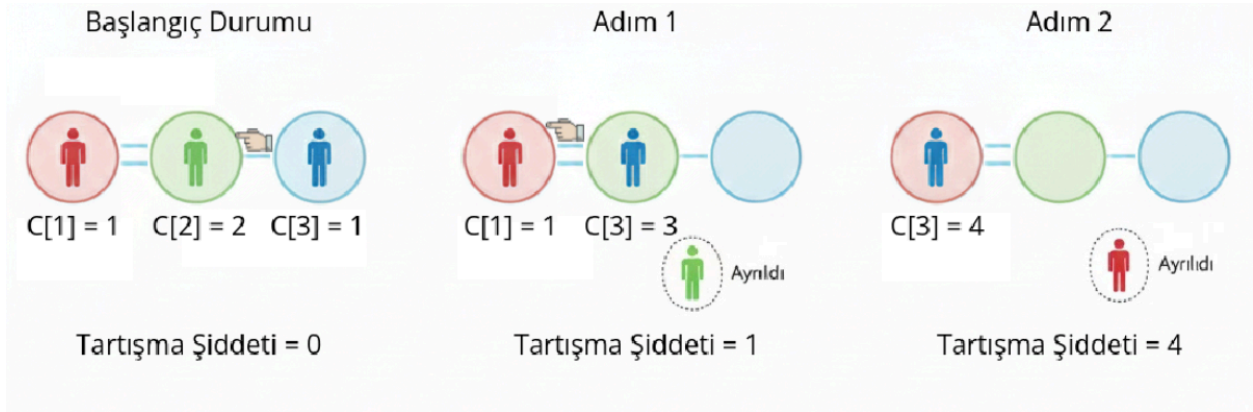
1  
3 3  
1 2 1  
1 2  
1 2  
2 3

## Örnek Çıktı

6 20

## Örnek Açıklama

Örnek bir tartışma aşağıdaki gibidir:



Ancak, örneğin bu örnek tartışma için Adım 1'den sonra 1 numaralı rehber 3 numaralı rehberi suçlayamaz. Bunun nedeni bir önceki suçlamayla aynı seviyede tartışma şiddetini artırmakta olmasına rağmen bu sefer suçlama yapan rehberin numarasının düşmesidir (bir önceki suçlamayı 3 numaralı rehber yapmışken şimdi 1 numaralı rehber yaptı).

Olası tüm tartışmalar şu şekilde listelenebilir:

1->2(1 numaralı yol), 1->3(3 numaralı yol). Tartışmanın toplam şiddeti:  $1+3=4$   
1->2(1 numaralı yol), 3->1(3 numaralı yol). Tartışmanın toplam şiddeti:  $1+1=2$   
1->2(2 numaralı yol), 1->3(3 numaralı yol). Tartışmanın toplam şiddeti:  $1+3=4$   
1->2(2 numaralı yol), 3->1(3 numaralı yol). Tartışmanın toplam şiddeti:  $1+1=2$   
3->2(3 numaralı yol), 3->1(1 numaralı yol). Tartışmanın toplam şiddeti:  $1+3=4$   
3->2(3 numaralı yol), 3->1(2 numaralı yol). Tartışmanın toplam şiddeti:  $1+3=4$

Toplam 6 farklı tartışma senaryosu var olup bu tartışmaların şiddetleri toplamı 20'dir.

# Antik Yazılar

**Zaman kısıtı:** 2 sn

**Memory kısıtı:** 256 MB

Antik yazılar üstüne yıllarını harcayan bilim insanları, bu yazıların bazı antik kelimeler içerdiğini farkettiler. Bir antik kelime birden fazla karakterden oluşan palindrom bir kelimeyi ardışık bir altdizi olarak içinde barındırmamalıdır. Örneğin, “kırkayak” kelimesi palindrom olan “kayak” kelimesini içerdiği için antik bir kelime sayılmaz. “ubilo” kelimesi ise antik bir kelimedir, çünkü palindrom bir kelime içermez. Bir kelimenin palindrom olmasının anlamı o kelimenin düz ve tersten okunuşunun aynı olmasıdır. Örneğin “kayak” kelimesi hem tersten hem düzden aynı olduğu için palindromken “metal” kelimesi tersten okunduğunda “latem” olduğu için palindrom değildir.

Sizden, verilen  $S$  antik yazısı üzerinde aşağıdaki sorguları gerçekleştirmeniz istenmektedir:

- $1\ i\ c$  - Verilen antik yazıdaki  $i$ . karakteri alfabenin  $c$ . harfi olarak değiştir. Yani  $C$  alfabenin  $c$ . harfi olmak üzere  $S_i = C$  olarak güncelle. Alfabe 26 karakterli İngiliz alfabesi olup harfler sırasıyla “abcdefghijklmnopqrstuvwxyz” şeklindedir. Örneğin bu alfabedeki 12. harf “l” harfidir. Bu güncellemenin ardından bütün yazının içinde kaç antik kelime olduğunu saymanız gerekmektedir. Diğer bir deyişle sizden  $1 \leq L \leq R \leq N$  olmak üzere kaç farklı  $S[L..R]$  kelimesinin antik olduğunu saymanız istenmektedir.
- $2\ l\ r$  - Verilen antik yazının  $l$  ile  $r$  karakterleri arasında kaç tane antik kelime olduğunu say. Diğer bir deyişle sizden  $l \leq L \leq R \leq r$  olmak üzere kaç farklı  $S[L..R]$  kelimesinin antik olduğunu saymanız istenmektedir.

## İmplementasyon Detayı:

Sizden istenen, aşağıdaki fonksiyonu implement etmenizdir. Program bir kere çalıştırıldığında fonksiyon birden fazla kere çağrılabilir.

```
antik (N, Q, S, T[], A[], B[])
```

Bu fonksiyon parametre olarak sırasıyla antik yazının uzunluğunu, antik yazıyı, sorgu sayısını, sorguların tipini ve içeriğini almaktadır. İçeriklerin tanımı şu şekildedir:

- $T[q] = 1$  ise bu sorgu  $i = A[q]$ ,  $c = B[q]$  parametreleriyle çağrılan ilk tip sorgudur. Bu durumda  $1 \leq B[q] \leq 26$  garanti edilmiştir.
- $T[q] = 2$  ise bu sorgu  $l = A[q]$ ,  $r = B[q]$  parametreleriyle çağrılan ikinci tip sorgudur. Bu durumda ise  $1 \leq A[q] \leq B[q] \leq N$  garanti edilmiştir.

Bu fonksiyon çıktı olarak  $Q$  elemanlı `long long` tipinde bir dizi dönmelidir. Bu dizinin her elemanı karşılık gelen sorgunun cevabını içermelidir. Daha fazla bilgi için verilmiş olan `antik-attachments.zip` içindeki dosyaları kontrol edebilirsiniz.

## Grader Kullanımı İçin Girdi ve Çıktı:

Girdinin ilk satırı `antik` fonksiyonunun kaç kere çağrılacağını, diğer bir deyişle test senaryolarının sayısını içerir.

Her test senaryosu için girdi formatı aşağıdaki gibidir:

```
N Q
S
T[1] A[1] B[1]
T[2] A[2] B[2]
...
T[Q] A[Q] B[Q]
```

Her test senaryosu için çıktı farklı satırlarda fonksiyonun döndüğü dizinin elemanı olan  $Q$  adet tamsayıyı içermektedir.

## Kısıtlar

- $\Sigma N$ ,  $\Sigma Q$  her fonksiyon çağrısı için değerlerin toplamı olsun.
- $1 \leq N, Q, \Sigma N, \Sigma Q \leq 2 * 10^5$
- $1 \leq T[q] \leq 2$
- $T[q] = 1$  ise  $1 \leq A[q] \leq N$ ,  $1 \leq B[q] \leq 26$
- $T[q] = 2$  ise  $1 \leq A[q] \leq B[q] \leq N$

## Alt Görevler

### Alt Görev 1 [3 puan]

$N, Q \leq 50$ ;  $\Sigma N, \Sigma Q \leq 10000$

### Alt Görev 2 [5 puan]

$N, Q \leq 200$ ;  $\Sigma N, \Sigma Q \leq 10000$

### Alt Görev 3 [9 puan]

$N, Q \leq 1000$

#### Alt Görev 4 [14 puan]

Alfabenin sadece ilk iki harfinin kullanıldığı garantidir. Yani  $S$  yazısı sadece “a” ve “b” harflerinden oluşmakta olup  $T[q] = 1$  ise  $1 \leq B[q] \leq 2$  garanti edilmiştir.

#### Alt Görev 5 [21 puan]

$$T[q] = 1$$

#### Alt Görev 6 [19 puan]

$$T[q] = 2$$

#### Alt Görev 7 [29 puan]

Ek kısıt yoktur.

#### Örnek Girdi

```
2
5 5
ubilo
2 1 5
1 3 15
2 1 3
1 2 2
2 2 4
4 2
abab
2 1 4
1 2 3
```

#### Örnek Çıktı

```
15
12
6
12
6
7
8
```

#### Örnek Açıklama

İlk senaryo için:

1..5 arası için geçerli kelimeler: {u, ub, ubi, ubil, ubilo, b, bi, bil, bilo, i, il, ilo, l, lo, o}.

Toplam: 15

Kelimenin 3. harfi alfabenin 15. harfi ile (o) değiştirildi. Yeni kelime: ubolo

1..5 arası için geçerli kelimeler: {u, ub, ubo, ubol, b, bo, bol, o, ol, l, lo, o}. Toplam: 12

1..3 arası için geçerli kelimeler: {u, ub, ubo, b, bo, o}. Toplam: 6

Kelimenin 2. harfi alfabenin 2. harfi ile (b) değiştirildi. Yeni kelime: ubolo

1..5 arası için geçerli kelimeler: {u, ub, ubo, ubol, b, bo, bol, o, ol, l, lo, o}. Toplam: 12

2..4 arası için geçerli kelimeler: {b, bo, bol, o, ol, l}. Toplam: 6

İkinci senaryo için:

1..4 arası için geçerli kelimeler: {a, ab, b, ba, a, ab, b}. Toplam: 7

Kelimenin 2. harfi alfabenin 3. harfi ile (c) değiştirildi. Yeni kelime: acab

1..4 arası için geçerli kelimeler: {a, ac, c, ca, cab, a, ab, b}. Toplam: 8

# Radyoaktivite Stabilizasyonu

**Zaman kısıtı:** 1.5 sn

**Memory kısıtı:** 256 MB

Radyoaktif bir maddeyi (bir string ile gösteriliyor) büyötmeye çalışıyorsunuz. Bu radyoaktif maddenin büyümesi  $n$  farklı adımdan oluşuyor. Her adımda, yeni bir harf geliyor (a, b ya da c). Her karakter geldikten sonra ise rastgele bir tepkime oluşuyor ve madde belli bir kuralı sağlamadan stabilize olmuyor. Bu kural ise şu şekilde:

- Eğer oluşan tepkime 0 ise maddedeki 'abc' şeklindeki ardışık alt dizilerin sayısının o uzunluk için maksimum değeri olması gerekiyor. (Örn: abcab stabil, ama aabbc stabil değil.)
- Eğer oluşan tepkime 1 ise maddedeki 'cba' şeklindeki ardışık alt dizilerin sayısının o uzunluk için maksimum değeri olması gerekiyor.

Bu tepkimeleri düzeltmek için herhangi bir karakteri alıp farklı bir karaktere dönüştürebilirsiniz. Her bir değıştirmenin size maliyeti 1 ise, bu büyötmeye operasyonunun tamamında (tüm stabilize işlemlerinin toplamı) elde edebileceğiniz minimum maliyet kaçır?

NOT: Her büyüme adımından sonra bir sonraki büyötmeye geçmeden maddeyi stabilize etmek zorunludur.

## Girdi:

Tek satırda  $s$  ve  $t$ .  $s_i$ ,  $i$ . adımda eklenen karakteri,  $t_i$  ise  $i$ . adımdaki eklenti sonucu oluşan tepkimenin tipini belirtiyor

## Çıktı:

Tek bir tamsayıda mümkün olan minimum maliyet

## Kısıtlar

Aşğıdaki gibi verilecektir.

- $1 \leq n \leq 100000$



## Alt Görevler

### Alt Görev 1 [9 puan]

$$1 \leq n \leq 20$$

### Alt Görev 2 [7 puan]

$t_i$  sabit (ya hep 1 ya hep 0)

### Alt Görev 3 [18 puan]

$$1 \leq n \leq 100$$

### Alt Görev 4 [16 puan]

$$1 \leq n \leq 500$$

### Alt Görev 5 [16 puan]

$$1 \leq n \leq 3000$$

### Alt Görev 6 [34 puan]

Ek kısıtlama yoktur.

## Örnek Girdi

abcba  
00101

## Örnek Çıktı

4

## Örnek Açıklama

a eklenir, stabil

b eklenir, stabil

c eklenir, cba sayısının maksimum olması için maddenin cba'ya dönüşmesi lazımdır, +2 dönüşüm.

b eklenir, abc sayısının maksimum olduğu seçeneklerden biri: abcb, +2.

a eklenir, abcba daki cba sayısı zaten maksimumdur.

# UbiloGPT

**Zaman kısıtı:** 3 sn

**Memory kısıtı:** 256 MB

Yaz kampında bütün Ulusal Bilgisayar Olimpiyatı öğrencileri olarak toplandınız ve iki hafta boyunca ikinci aşamaya çalışmak yerine piyasadaki bütün LLM'lerden daha iyi olacak bir yapay zeka ürettiniz. Bu yapay zekanın adını da tabii ki UbiloGPT koydunuz.



(Tabii ki olimpiyatçılar arasında hiç kimsenin tasarım geçmişi olmadığı için logoyu tasarlayacak Nano Ubilo Pro adında başka bir yapay zeka da ürettiniz.)

Şimdi sırada yapay zekanın seviyesini ölçmek var. Bunun için UbiloGPT'ye  $N$  adet  $K$  karakterli **rastgele oluşturulmuş** isimler içeren bir dizi verdiniz ve bu dizi hakkında  $Q$  adet sorgu sormaya karar verdiniz. Her bir sorguda UbiloGPT'ye  $K$  karakterden oluşan  $A$  ve  $B$  isimleri verdiniz ve ondan dizideki bütün alt dizilerden kaç tanesindeki bütün isimlerin alfabetik olarak  $A$  ile  $B$  arasında olduğunu cevaplamasını istediniz.

UbiloGPT'nin cevaplarının doğruluğunu kontrol etmek için verilen sorguların gerçek cevaplarını hesaplayacak bir program yazabilir misiniz?

## Girdi

İlk satırda iki adet tam sayı,  $N$  ve  $Q$ . İsim ve sorgu sayıları.

İkinci satırda  $N$  adet  $K$  karakterden oluşan bir isim dizisi. UbiloGPT'ye verilen dizi. İsimlerin her harfi A-Z arasından rastgele seçilmiştir.

Sonraki  $1 \leq i \leq Q$  için  $i$ . satırda iki adet  $K$  karakterden oluşan  $A_i$  ve  $B_i$  isimleri. Verilen sorunun alfabetik olarak alt ve üst limiti.  $A_i$  ve  $B_i$  rastgele oluşturulmuş olmak zorunda değildir.

## Çıktı

$1 \leq i \leq Q$  için  $i$ . satırda içindeki bütün isimlerin alfabetik olarak  $A_i$  ile  $B_i$  arasında olduğu dizi sayısı.

## Kısıtlar

- $1 \leq N, Q \leq 3 * 10^5$
- $1 \leq K \leq 10$
- $A_i \leq B_i$

## Alt Görevler

### Alt Görev 1 [7 puan]

- $N, Q \leq 50$

### Alt Görev 2 [9 puan]

- $N, Q \leq 3000$

### Alt Görev 3 [13 puan]

- Tüm  $A_i$  isimleri tamamen A harflerinden oluşmaktadır.

### Alt Görev 4 [11 puan]

- $K = 1$

### Alt Görev 5 [19 puan]

- $N \leq 1000$

### Alt Görev 6 [41 puan]

- Ek Kısıt Yoktur

## Örnek Girdi 1

```
6 3
QWE ZXC ASD ERT ASD SDF
ABC FGH
QRT ZXC
AAA ZZZ
```

## Örnek Çıktı 1

```
6
4
21
```

## Örnek Açıklama 1

İlk sorgu için sadece ABC ile FGH arasında isimler içeren alt diziler:

[ASD], [ERT], [ASD], [ASD, ERT], [ERT, ASD], [ASD, ERT, ASD]

İkinci sorgu için sadece QRT ile ZXC arasında isimler içeren alt diziler:

[QWE], [ZXC], [SDF], [QWE, ZXC]

## Örnek Girdi 2

```
10 5
OPUNX NJJQM XZGHT DSCQD XOFZM TUTMJ BJUCS YVSBW FVLUY GHYEV
TXTZH VDGRA
EUQDF WOFDL
RKLWD STDVH
CLXMH ZWHYV
KYBQK WPKIQ
```

## Örnek Çıktı 2

```
0
7
0
27
4
```

### Örnek Girdi 3

```
10 5
S H I O K M U T Y E
I T
C P
E Y
A W
L Q
```

### Örnek Çıktı 3

```
12
16
55
37
2
```

# Mangal

**Zaman limiti:** 3 sn

**Memory limiti:** 256 MB

Öğrenciler için kampların en önemli anlarından birisi mangal günüdür. Yine bir gün mangal hazırlanırken ızgaranın sıcaklığının ayarlanmasında sorunlar yaşandığını fark ettiniz ve olimpiyat yeteneklerinizi kullanarak bunu çözmek için bir program yazmaya karar verdiniz.



Mangal ızgarası  $N$  satır ve  $M$  sütundan oluşmaktadır ve her hücresi ilk başta 0 derecedir. Siz herhangi bir operasyonda ya ızgaranın bütün bir satırının sıcaklığını 1 derece arttırabilir ya da bütün bir sütununun sıcaklığını 1 derece azaltabilirsiniz.

Oldukça farklı türde yemekler pişirileceği için her bir hücrenin kabul edilebilir bir sıcaklık aralığı vardır. Eğer bir hücrenin sıcaklığı  $H_{i,j}$  ise, verilen  $A_{i,j}$  ve  $B_{i,j}$  değerleri için  $A_{i,j} \leq H_{i,j} \leq B_{i,j}$  sağlanmalıdır.

Eğer sonlu sayıda operasyonlarla bu sıcaklık koşullarını sağlayan bir ızgara oluşturmak mümkünse o ızgaralardan herhangi birinin her bir hücresinin sıcaklığını yazdırın. Eğer bunu başarmak imkansız ise de bunu belirtin.

Ayrıca, kamp süresince birden fazla mangal günü yapılacağı için bu hesaplamaları  $T$  farklı gün için yapmanız gerekiyor.

## Girdi

İlk satırda bir adet tam sayı,  $T$ . Kamptaki mangal günü sayısı.

Her mangal günü için ilk satırda iki adet tam sayı,  $N$  ve  $M$ . O günkü ızgaranın satır ve sütun sayısı.

Takip eden  $1 \leq i \leq N$  için  $i$ . satırda  $1 \leq j \leq M$  için  $A_{i,j}$  tam sayıları. Hücrelerin alt sıcaklık limitleri.

Takip eden  $1 \leq i \leq N$  için  $i$ . satırda  $1 \leq j \leq M$  için  $B_{i,j}$  tam sayıları. Hücrelerin üst sıcaklık limitleri.

## Çıktı

$1 \leq i \leq T$  için eğer  $i$ . günün koşullarına uygun bir sıcaklık ayarı mümkünse EVET yazdırıp ızgaranın her bir hücrelerinin sıcaklıklarını sonraki  $N$  satırda  $M$  adet tam sayı yazdırarak belirtin. Bunu başarmak imkansız ise HAYIR yazdırın.

## Kısıtlar

- $1 \leq T \leq 10$
- $1 \leq N, M \leq 300$
- $-2 * 10^9 \leq A_{i,j} \leq B_{i,j} \leq 2 * 10^9$

## Alt Görevler

### Alt Görev 1 [11 puan]

- $A_{i,j} = B_{i,j}$

### Alt Görev 2 [14 puan]

- $A_{i,j} = B_{i,j}$  eğer  $i = 1$  ya da  $j = 1$  ise

### Alt Görev 3 [16 puan]

- $N, M \leq 9$  ve  $B_{i,j} - A_{i,j} \leq 1$

### Alt Görev 4 [17 puan]

- $A_{i,j} = B_{i,j}$  eğer  $i + j = 0 \pmod{2}$  ise

### Alt Görev 5 [42 puan]

- Ek Kısıt Yoktur

## Örnek Girdi 1

```
2
2 2
1 -2
0 -3
4 0
1 -2
2 2
0 0
0 1
0 0
0 1
```

## Örnek Çıktı 1

```
EVET
1 -2
1 -2
HAYIR
```

## Örnek Açıklama 1

İlk gün için eğer her satırın sıcaklığı 1 artırıp ikinci sütunun sıcaklığını 3 azaltırsanız verilen sıcaklık değerlerine ulaşırsınız ve bu sıcaklık değerleri verilen koşulları sağlamaktadır. Fakat tek çözüm bu olmak zorunda değildir. Örneğin ilk satırın sıcaklığını 1 yerine 2 kere de arttırabilirsiniz.

İkinci gün için verilen operasyonlarla istenilen sıcaklık koşullarının sağlanamayacağı ispatlanabilir.



# XORting

**Zaman kısıtı:** 2 sn

**Memory kısıtı:** 512 MB

Sirtok, Özköse ve Ersun, tren kaçırmaktan hoşlaşan üç arkadaşlardır. Zonguldak'tan kaçıracakları treni gözlerine kestirmiş ve olabildiğince iyi hazırlanmış olsalar da, erteleme alışkanlıkları sebebiyle bir anda kendilerini trenin üstünde bulmuşlardır. Düldül ismindeki bu tren,  $N$  vagondan oluşmaktadır. Üç arkadaş, hazırlıklarını iyi yaptıkları için her bir vagonun içindekilerin ekonomik değerini kuruşu kuruşuna bilmektedirler, ve bu değerleri kendi aralarında  $C_i$  dizisi ile gösterirler. Bazı vagonlar boş olabilmektedir, bazıları dört tane müzisyen ile bir tanesinin büyükbabasını taşıyabilir, bazılarında ise Linus Pauling'in sonraki dersini mutlu bir şekilde işleyebilmesi için kilogramlarca asbestos ve C vitamini tableti vardır.

Maalesef Ersun, komik olacağını düşündüğü için, planlarının tam ortasında makiniste aynadan görebileceği şekilde nanik yapar. Bunu gören makinist, bütün planı anlar ve güvenlik güçlerine haber vermek için elini olabildiğince çabuk tutmaya yeltenir. Durumlarının vahimliğini anlayan Sirtok:

*Yakalanırsak çok pis XORtlarız, olamaz, hayallerim suya düşüyor!*

şeklinde hayıflanır. Kaçmak için 2 saniyeleri olsa ve bu süre fazlasıyla yeterli olsa da, Özköse sonraki hafta İstanbul'da vereceği olimpiyat dersini düşündüğü için Sirtok'ın yakınısını kafasında bir probleme dönüştürür. Kafasından geçenler şöyledir: Bir vagon tamamen jelibon dolu olsa, değeri  $J$  olurdu. Biz yakalanmasaydık, istediğimiz ardışık vagonlar dizisini çalabilirdik. Bu ardışık dizideki her bir vagonun karizması, kendisi dahil kendisinden sonraki bütün çalınan vagonların değerlerinin XOR'u olsa, *Jelibon karşılaştırması*'na göre çalabileceğimiz en küçük değerli vagon karizma altdizisi hangisi olurdu?

Maalesef Özköse'nin dikkatı dağıldığı için bulundukları zor durumu tamamen unutup bir çözüm üstüne düşünmeye başladı. Üçlümünün problemini süre sınırları içerisinde çözerek Özköse'nin dikkatini toplayıp kaçmayı hatırlamasına yardım etmek size düşüyor. *Jelibon karşılaştırması*'nın tanımı, ve sizden istenilen değerler sonraki sayfada detaylı olarak verilmiştir. (Eğer hala kafa karışıklığınız varsa, Ek'teki ("*extra/helper.h*") karşılaştırma ve değer hesaplama için verilen fonksiyonlara da bakabilirsiniz.) Eğer birden fazla cevap varsa, herhangi birini bulabilirsiniz.

*Jelibon karşılaştırması*, iki dizi arasında, kısa olanın sonuna, uzunlukları eşit olana dek  $J$  sayısı eklendikten/bağlandıktan sonra yapılan standart alfabetik karşılaştırma şeklinde tanımlanmıştır. Eğer, uzunluk eşitleme işlemi sonucunda diziler alfabetik karşılaştırmaya göre aynıysa, **orijinal olarak uzun olan dizi** daha küçük olarak alınır. (Bu karşılaştırmanın, sorunsuz bir sıralama ilişkisi tanımladığı kanıtlanabilir, yani hiçbir  $A, B, C$  dizi üçlüsü, *Jelibon karşılaştırması*'na göre  $A < B$ ;  $B < C$ ;  $C < A$  koşullarının üçünü de aynı anda sağlamaz.)

Bir vagon aralığı  $[L, R)$  için (sol sınırı dahil, sağ sınırı dahil değil),  $R - L$  uzunluğundaki 0 indeksli karizma dizisi aşağıdaki şekilde tanımlanmıştır.

$$B_{L,R,i \in [0,R-L)} = \bigoplus_{j=i+L}^{R-1} C_j$$

Küçüklük operatörünün *Jelibon karşılaştırması*'na göre tanımlandığı varsayılırsa, sizden istenen aşağıdaki koşulu sağlayan  $L, R$  ikililerinden herhangi birini bulmanızdır.

$$\forall 0 \leq l < r \leq N \quad B_{l,r} \not\prec B_{L,R}$$

## İmplementasyon Detayı

Size verilen aşağıdaki fonksiyonu implement etmeniz gerekmektedir.

`(L, R) xort (C[], J)`

- $C$ , vagonlardaki eşyaların değerlerini temsil eden dizidir.
- $J$ , jelibon dolu bir vagonun değerini temsil eden bir tamsayıdır.
- Bu fonksiyonun, bir aralığı temsil eden iki tamsayı (sınır dahil başlangıç indeksi ve sınır dahil olmadan bitiş indeksi) döndürmesi gerekmektedir.

Bu fonksiyon her bir testte  $I$  defa çağırılacaktır. Global değişkenlerinizin önceki fonksiyon çağrıları sebebiyle değişebileceğini unutmayın.

Girdi ve çıktılar size verilen *grader.cpp* tarafından yapılacaktır, çözüm kodunuzda standart girdi ya da çıktıyla herhangi bir şekilde etkileşim olursa (cin, cout, printf, scanf) çalışma zamanı ya da "Protocol Violation" hatası alacaktır veya yanlış sayılacaktır. Standart hata çıktısını (cerr, fprintf(stderr, ...)) istediğiniz şekilde kullanabilirsiniz.

## Girdi:

İlk satırda  $I$  (test sayısı) olmalıdır. Sonrasında, her bir test aşağıdaki formatta olmalıdır.

```

$$\begin{matrix} N & J \\ C_0 & C_1 \cdots C_{N-1} \end{matrix}$$

```

## Çıktı:

$I$  testin her biri için döndürdüğünüz cevap aşağıdaki formatta verilecektir.

```

$$L \ R$$

```

- $L$  döndürdüğünüz ilk sayı,  $R$  ise ikincisidir. (Ek olarak size verilen *grader.cpp*, bu sayıların bir altdiziyi temsil edip etmediğini kontrol etmektedir. CMS'te değerlendiren kodların, döndürülen cevap bir altdiziyi temsil etmiyorsa çalışma zamanı hatası **vermeyeceğini** unutmayın. )

## Kısıtlar

- $1 \leq I \leq 10^6$
- $\sum N$ , değişkenin  $I$  testteki değerlerinin toplamını temsil etmektedir.
- $N > 0$
- $0 \leq C_i < 2^{30}$
- $-1 \leq J \leq 2^{30}$
- $\sum N \leq 10^6$
- Bütün verilen sayılar tamsayılardır. 🐎

## Alt Görevler

\* Yanlarında bağlı oldukları (puan almanız için geçmesi gereken) diğer altgörevler yazmaktadır.

### Alt Görev 1 [4 puan]

$C_i < 2; \sum N \leq 500$ ; Dizideki her sayı bir değerinden bağımsız üretilmiştir.

### Alt Görev 2 [7 puan] [1]

$C_i < 2; \sum N \leq 2000$ ; Dizideki her sayı bir değerinden bağımsız üretilmiştir.

### Alt Görev 3 [4 puan]

$C_i < 2; \sum N \leq 3 \cdot 10^5; J = -1$

### Alt Görev 4 [7 puan]

$C_i < 2; \sum N \leq 3 \cdot 10^5; J = 0$

### Alt Görev 5 [9 puan] [1][2][3][4]

$C_i < 2; \sum N \leq 3 \cdot 10^5$ ;

### Alt Görev 6 [8 puan] [1]

$\sum N \leq 500$

### Alt Görev 7 [8 puan] [1][2][6]

$\sum N \leq 2000$

### Alt Görev 8 [9 puan] [3]

$J \in \{-1, 2^{30}\}; \sum N \leq 3 \cdot 10^5$

### Alt Görev 9 [22 puan] [1][2][3][4][5][6][7][8]

$\sum N \leq 3 \cdot 10^5$

### Alt Görev 10 [6 puan]

$\exists A_i = 0$  (Dizde en az bir adet 0 olacağı garanti edilmiştir.)

## Alt Görev 11 [16 puan] [1][2][3][4][5][6][7][8][9][10]

Ek kısıt yoktur.

### Örnek Girdi

```
4
5 -11
1 2 3 2 1
9 8
1 2 3 1 2 3 1 2 3
10 0
5 7 4 5 3 1 2 4 3 2
10 0
0 1 0 1 0 2 0 2 0 2
```

### Örnek Çıktı

```
0 3
0 9
4 7
8 9
```

### Örnek Açıklama

Size verilen karşılaştırma fonksiyonlarıyla örnek çıktıları doğrulayıp örnek girdiyi daha iyi anlamaya çalışabilirsiniz. Ek dosyasında size bir örnek girdi daha verilmiştir. Bu girdi ( $02.in$ ),  $C_i < 2$  koşulunu sağlamaktadır.