

Okul Kulüpleri

Zaman kısıtı: 2 sn

Hafıza kısıtı: 128 MB

Bir sosyal bilimler lisesinde hafta içi dersler tamamlandıktan sonra öğrenci kulüpleri faaliyetlerine başlayacaktır. Okul müdürü Engin Bey, önümüzdeki dönemin planlamasını yapmak için şimdiden her bir öğrenciye hangi kulüpler ile ilgilendiğini sormayı planlar. Okulda toplam N tane öğrenci ve M tane öğrenci kulübü vardır. Her öğrenci ilgi duyduğu toplam 5 tane kulübü bildirecektir. Kulüpler pozitif tam sayı ile belirtilir.

Okul imkanlarını mümkün olduğunca verimli kullanmayı planlayan Engin Bey, tercihleri farklı olan öğrenci çiftlerinin sayısını bulmak ister. Bir öğrenci çiftinin, yani öğrenci A ve öğrenci B 'nin tercihleri farklıdır eğer bu iki öğrencinin hiç bir okul kulübü tercihi aynı değil ise.

Sosyal bilimler lisesinde kodlama ile ilgilenen öğrenci olmadığından harika kodlama ve algoritma yeteneklerine sahip bilgisayar olimpiyatçısı olarak sizin Engin Bey'e yardımcı olmanız istenmektedir.

Girdi Formatı

Girdinin ilk satırında öğrenci sayısı (N) vardır.

Takip eden N satırda, her biri farklı bir öğrencinin 5 farklı kulüp tercihini içeren bilgiler yer almaktadır. Her satırdaki 5 tam sayı, öğrencinin ilgi duyduğu kulüpleri temsil etmektedir ve bu sayılar birbirinden farklıdır.

Çıktı Formatı

Tercihleri farklı olan öğrenci çiftlerinin sayısını tek bir tam sayı olarak yazdırın.

Limitler

- $2 \leq N \leq 50.000$
- $1 \leq M \leq 10^6$

Alt Görevler

Alt Görev 1 (15 Puan)

- $2 \leq N \leq 10$

Alt Görev 2 (30 Puan)

- $2 \leq N \leq 20.000$

Alt Görev 3 (55 Puan)

- Ek kısıt yoktur.

Örnek Girdi

```
3
1 2 3 4 5
6 7 8 9 10
11 12 13 14 1
```

Örnek Çıktı

```
2
```

Örnek Açıklaması

İki öğrenci çiftinin tercihleri farklıdır. Daha açık olarak, birinci ve ikinci öğrencinin ve ayrıca, ikinci ve üçüncü öğrencinin tercihleri farklıdır. Bu nedenle çıktı olarak 2 verilir.

Olimpik Seçim

Zaman kısıtı: 3 sn

Memory kısıtı: 256 MB

Olimpiyat sisteminin geliştirilmesi için bu yılın olimpiyatlarında farklı bir sistem uygulanmıştır. Her bir olimpiyatçıya önem sırasına göre K adet test yapılmıştır ve bu testler skorlandırılmıştır. Bu testlere göre bir olimpiyatçının diğer bir olimpiyatçıdan daha başarılı sayılması için ilk testte daha yüksek puan alması, ilkinde aynı puanı aldığı takdirde ikincisinde daha yüksek puan alması, ikincisinde aynı puanı aldığı takdirde üçüncüsünde daha yüksek puan alması vs. gerekmektedir. Eğer iki olimpiyatçı bütün testlerden eşit puan aldıysa o zaman o iki olimpiyatçının başarıları eşdeğer sayılmaktadır.

Olimpiyatçılar testlere sırayla girmektedir ve skor bilgileri size aynı sırayla verilmektedir. Sizden istenen bir olimpiyatçının skor bilgisini aldıktan sonra o ana kadarki olimpiyatçılar arasından en yeni olimpiyatçıyı da içerecek şekilde seçilebilecek maksimum olimpiyatçı sayısını bildirmeniz.

M adet olimpiyatçının seçildiği bir seçimi $S_1, S_2, \dots, S_{M-1}, S_M$ dizisiyle gösterebiliriz. S_i seçilen i . olimpiyatçının kaçınıcı sırada testlere girdiğini belirtmektedir. Bu dizi şu kuralları sağlamalıdır:

1. $S_i < S_{i+1}$ yani dizideki olimpiyatçıların sırası o olimpiyatçıların testlere girme sıralarıyla aynı olmalıdır.
2. $S_{i+1} - S_i \leq D$ yani dizideki ardışık iki olimpiyatçının arasında testlere girmiş olan D veya daha fazla olimpiyatçı olamaz.
3. S_{i+1} olimpiyatçısı S_i olimpiyatçısından daha başarılı olmalıdır. Yani seçilen olimpiyatçılar başarılarına göre artan sırada olmalıdır.

İmplementasyon Detayları

İki adet fonksiyon yazmanız gerekiyor:

```
void baslangic(int N, int K, int D)
```

- N : Testlere girecek olimpiyatçı sayısı.
- K : Test sayısı.

- D : Dizi kurallarında tanımlanan değer.
- Bu fonksiyon başlangıçta hiçbir fonksiyon çağırılmamışken sadece bir kez çağırılacaktır.

```
int yeni_olimpiyatci(int[] P)
```

- P : Yeni olimpiyatçının testlerden aldığı skorları belirten K boyutunda dizi. $P[i]$ değeri i testindeki skoru belirtir.
- Bu fonksiyon o ana kadar testlere giren olimpiyatçıların arasından en yeni olimpiyatçıyı da içeren kurallara uygun bir seçimdeki maksimum olimpiyatçı sayısını döndürmelidir.
- Bu fonksiyon tam olarak N kere çağırılacaktır.

Örnekler

Bu fonksiyon çağırmasını düşünün:

```
baslangic(4, 2, 2)
```

Bu fonksiyon 5 adet olimpiyatçının testlere gireceğini, her olimpiyatçının 2 adet teste gireceğini ve seçilen iki olimpiyatçının arasında testlere girmiş 2 veya daha fazla olimpiyatçı olamayacağını belirtmektedir.

```
yeni_olimpiyatci([1, 1])
```

Bu fonksiyon 1 döndürmelidir çünkü seçilebilecek sadece 1 olimpiyatçı vardır.

```
yeni_olimpiyatci([3, 2])
```

Bu fonksiyon 2 döndürmelidir çünkü uygun seçimler $[2]$, $[1, 2]$ 'dir. $[1]$ uygun bir seçim değildir çünkü en yeni olimpiyatçıyı içermemektedir.

```
yeni_olimpiyatci([3, 1])
```

Bu fonksiyon 2 döndürmelidir çünkü uygun seçimler [3], [1, 3]'dir. [2, 3] ve [1, 2, 3] seçimleri uygun seçimler değildir çünkü 3. olimpiyatçı 2. olimpiyatçıdan daha başarılı değildir.

```
yeni_olimpiyatci([2, 2])
```

Bu fonksiyon 1 döndürmelidir çünkü tek uygun seçim [4] seçimidir. [1, 4] uygun bir seçim değildir çünkü 1. ile 4. olimpiyatçı arasında 2 olimpiyatçı vardır ve bu 2. kurala uymamaktadır.

Kısıtlar

- $1 \leq N, K, N \times K \leq 10^6$
- $0 \leq D < N$
- $0 \leq P[i] \leq 10^9$

Alt Görevler

Alt Görev 1 [4 puan]

- $N \leq 20$
- $K = 1$
- $D = N - 1$
- $P[i] < N$

Alt Görev 2 [7 puan]

- $N \leq 10^3$
- $K = 1$
- $D = N - 1$
- $P[i] < N$

Alt Görev 3 [12 puan]

- $K = 1$
- $D = N - 1$
- $P[i] < N$

Alt Görev 4 [15 puan]

- $K = 1$
- $D = N - 1$

Alt Görev 5 [9 puan]

- $K = 1$
- $P[i] < N$

Alt Görev 6 [18 puan]

- $D = N - 1$
- $P[i] < N$

Alt Görev 7 [35 puan]

- Ek Kısıt Yoktur

Örnek Çalıştırıcı

Kodunuzu test etmek için kullanabileceğiniz örnek çalıştırıcı (grader) girdiyi şu şekilde almaktadır:

- Satır 1: $N K D$
- Satır 2 + i ($0 \leq i \leq N - 1$): $P[0] P[1] P[2] \dots P[K - 2] P[K - 1]$

Ayrıca şu şekilde çıktı vermektedir:

- Satır 1 + i ($0 \leq i \leq N - 1$): Her yeni_olimpiyatci fonksiyonunun döndürdüğü cevap.

RoboX

Zaman kısıtı: 1 sn

Memory kısıtı: 128 MB

Kalabalık Teknovil şehrinde, RoboX adlı bir teslimat robotuna önemli paketlerin çeşitli depolar arasında taşınması görevi veriliyor. Şehir, N depodan oluşur ve RoboX, depo 1'den başlayıp depo N 'ye kadar olan her depoyu sırayla ziyaret etmelidir.

Ancak RoboX, teslimat rotasını optimize etmek için K adet depoya kadar atlamasına olanak tanıyan gelişmiş bir navigasyon sistemi ile donatılmıştır. Teslimatta ilk ve son depoların, yani depo 1'in ve depo N 'nin atlanmasına izin verilmez.

Amacınız, 0 ile K arasında (sınırlar dahil) depo atlama seçeneği göz önüne alındığında, RoboX'un alması gereken minimum mesafeyi bulmaktır. Burada (x_1, y_1) ve (x_2, y_2) lokasyonlarındaki iki depo arasındaki mesafe $|x_1-x_2| + |y_1-y_2|$ ile hesaplanır.

Girdi Formatı

Girdinin ilk satırı N ve K tam sayılarını içerir.

Ardından depoların koordinatlarını gösteren N tane x,y tamsayı çifti verilir.

Çıktı Formatı

Tek bir tamsayı çıktısı verin. Bu tamsayı RoboX'un en fazla K depoyu atlamasını içerecek şekilde teslimat rotasını tamamlamak için seyahat etmesi gereken minimum mesafeyi temsil eder.

Limitler

- $3 \leq N \leq 500$
- $-1000 \leq x, y \leq 1000$
- $K < N$

Alt Görevler

Alt Görev 1 (9 Puan)

- $N \leq 10$

Alt Görev 2 (37 Puan)

- $N \leq 100$

Alt Görev 3 (54 Puan)

- Ek kısıt yoktur.

Örnek Girdi

```
6 3
2 2
9 4
3 3
11 -8
20 30
4 4
```

Örnek Çıktı

```
4
```

Örnek Açıklaması

En fazla 3 tane depo atlanabilir. (9,4), (11 -8), ve (20,30) koordinatlarındaki depoların atlanması sonrası RoboX (2,2), (3,3) ve (4,4) lokasyonlarındaki depoları ziyaret eder. Minimum mesafa bu durumda 4 olur.

Sayılar Diyarı

Zaman kısıtı: 3 sn

Memory kısıtı: 256 MB

Büyücülerin yaygın olduğu Sayılar Diyarında her büyücüye doğuştan bir sayı atanır. Bir büyücü için bu sayı çok değerlidir çünkü o büyücünün neleri yapabileceğini belirtir.

Örneğin bu diyardaki her bir yolun bir sayı değeri vardır ve bir büyücünün bir yolu kullanabilmesi için o büyücünün sayısının bölenlerinin toplamının yolun sayısının bölenlerinin toplamından büyük olması gerekir.

Sizden istenen size verilecek olan Q adet sorguyu cevaplamanız. Her sorguda bir büyücünün bulunduğu şehir ile o büyücünün sayısı verilecektir ve o büyücünün kaç farklı şehire ulaşabileceği hesaplamanız istenmektedir.

Bunu yapabilmemiz için size bu diyardaki N şehir arasındaki çift yönlü M yolun başlangıç ve bitiş şehirleri ile sayı değerleri verilecektir. Başlangıç ve bitiş noktası aynı şehir olan yollar olabilir, iki şehir arasında birden fazla yol olabilir ve bütün şehirlerin birbirine ulaşım sağlayabileceğinin garantisi yoktur.

İmplementasyon Detayları

İki adet fonksiyon yazmanız gerekiyor:

```
void baslangic(int N, int M, int[] A, int[] B, int[] C)
```

- N : Şehir sayısı.
- M : Yol sayısı.
- A, B, C : Yolları temsil eden M boyutunda diziler. $(A[i], B[i], C[i])$ değerleri sırasıyla i . yolun başlangıç şehrini, bitiş şehrini ve sayısını belirtir.
- Bu fonksiyon diğer fonksiyonlardan önce ve bir kez çağırılacaktır.

```
int sorgu(int V, int K)
```

- V : Büyücünün bulunduğu şehir.
- K : Büyücünün sayısı.
- Bu fonksiyon V şehrinden başlayan bir büyücünün ulaşabileceği şehir sayısını döndürmelidir.

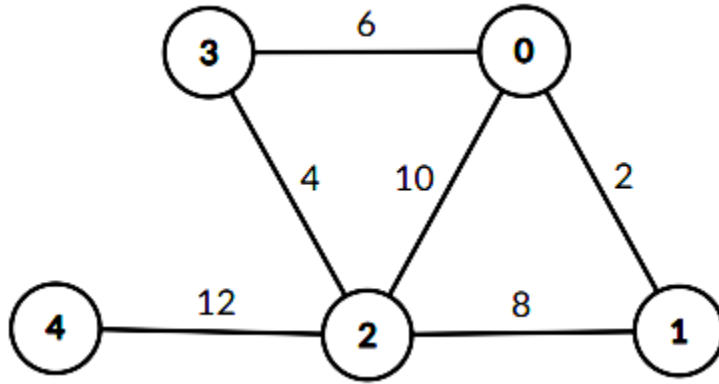
- Bu fonksiyon tam olarak Q kere çağırılacaktır.

Örnekler

Bu fonksiyon çağrısını düşünün:

```
baslangic(5, 6, [0, 0, 0, 1, 2, 2], [1, 2, 3, 2, 3, 4],  
[1, 10, 6, 8, 4, 12])
```

Bu çağrıda verilen bilgiler aşağıdaki graf şeklinde gösterilebilir.



Sonradan yapılacak sorgu çağrıları şunlar olabilir:

```
sorgu(1, 14)
```

Bu sorgu sonuç olarak 4 döndürmelidir çünkü 1 şehrinden başlayan 14 sayılı bir büyücü sadece 0, 1, 2, 3 şehirlerine ulaşabilir.

```
sorgu(2, 16)
```

Bu sorgu sonuç olarak 5 döndürmelidir çünkü 2 şehrinden başlayan 16 sayılı bir büyücü bütün şehirlere ulaşabilir.

Kısıtlar

- $1 \leq N \leq 10^5$
- $1 \leq Q, M \leq 5 \times 10^5$
- $0 \leq A[i], B[i] \leq N - 1$ ve $1 \leq C[i] \leq 3 \times 10^7$ (bütün $0 \leq i \leq M - 1$ için)
- $0 \leq V[i] \leq N - 1$ (bütün $0 \leq i \leq Q - 1$ için)
- $1 \leq K[i] \leq 3 \times 10^7$ (bütün $0 \leq i \leq Q - 1$ için)

Alt Görevler

Alt Görev 1 [5 puan]

- $N, M, Q \leq 10^2$
- $C[i], K[i] \leq 10^3$

Alt Görev 2 [10 puan]

- $N, M, Q \leq 10^3$

Alt Görev 3 [25 puan]

- $Q \leq 25$

Alt Görev 4 [35 puan]

- $C[i], K[i] \leq 10^3$

Alt Görev 5 [25 puan]

- Ek Kısıt Yoktur

Örnek Çalıştırıcı

Kodunuzu test etmek için kullanabileceğiniz örnek çalıştırıcı (grader) girdiyi şu şekilde almaktadır:

- Satır 1: $N M Q$
- Satır 2 + i ($0 \leq i \leq M - 1$): $A[i], B[i], C[i]$
- Satır 2 + $M + i$ ($0 \leq i \leq Q - 1$): $V[i], K[i]$

Ayrıca şu şekilde çıktı vermektedir:

- Satır 1 + i ($0 \leq i \leq Q - 1$): Sorgu i için sorgu fonksiyonunun döndürdüğü cevap.

Fırtına

Zaman Kısıtı: 2sn

Hafıza Kısıtı: 128 MB

Önümüz kış! Bir fırtına yaklaşıyor. Seninle bulunduğumuz ülkenin haritasını paylaşıyorum. Haritada toplam N adet şehir var ve bu şehirler M adet yol ile birbirine bağlanmış durumda. Meteoroloji kuvvetli bir yağış ön görüyor böyle giderse şehirler arası yollar kullanılmaz olacak. Bu yüzden öncesinde şehirlere ulaşım gerekli tahkimatın (güçlendirme) yapılması gerekiyor. Umarım tüm şehirleri güçlendirecek kadar vaktimiz vardır. Ama yine de: hedefimiz maksimum sayıda şehirde fırtınaya karşı önlem almak.

Afet birimleri her bir şehirde tahkimat için ne kadar zaman harcanması gerektiğini hesapladı. Bir u şehirde toplam t_u zaman geçirip gerekli tüm güçlendirmeyi yapabiliyoruz. Asıl vakit kaybettiren kısmın güçlendirme olduğunu düşündüğümüz için iki şehir arasında gitmenin hiç vakit kaybettirmediğini varsayıyoruz. Ancak, meteoroloji her yolun hangi saatten itibaren kullanılmaz olacağını hesapladı. Herhangi u ve v şehirlerini bağlayan bir e yolu f_e zamanında sonra kullanılmaz hale gelecek. Unutma f_e anında hala yolu kullanabilirler ancak sonrasında yol kullanıma kapatılacak. Başlangıç zamanı 0 dır.

Önlemlere dair işlemleri gerçekleştirebilecek yetkinliğe sahip sadece bir ekip var ve 1 numaralı şehirde bulunuyor. Ayrıca bu ekibin tüm işleri bitirdiğinde yine 1 numaralı şehirde bulunması gerekiyor. Senden istenen bu ekip için en ideal iş sırasını oluşturup maksimum sayıda şehirde tahkimat yapılmasını sağlaman. Verilen harita, meteoroloji verisi ve afet birimleri öngörülerini ışığında; çıktıda senden, tek satırda en fazla kaç şehir güçlendirme yapılabilecek bir iş sırası olduğunu belirten tek bir tamsayı istiyoruz.

Girdi Formatı

İlk satırda N ve M değerlerini belirten iki tam sayı yer alacak.

İkinci satırda her şehir için tahkimat zamanlarını belirten N adet tam sayı t_i . Bu t_i tam sayısı i numaralı şehri güçlendirmek için gereken süreyi göstermektedir.

Sıradaki M satırda yollar ve hangi saatte kullanıma kapatılacağını belirten üç adet tam sayı yer alacaktır. Sırasıyla u , v ve f tam sayıları u ile v arasında bir yol bulunduğunu ve f anında kullanıma kapatılacağını belirtmektedir.

Çıktı Formatı

Tek satırda en fazla kaç şehirde tahkimat yapılabileceğini belirten tek bir tam sayı.

Limitler

- $1 \leq N \leq 100\ 000$
- $1 \leq M \leq 500\ 000$
- $0 \leq f_e \leq 10\ 000$
- $0 \leq t_u \leq 10\ 000$

Alt Görevler

Alt Görev 1 (17 Puan)

- $1 \leq N \leq 10$
- $1 \leq M \leq 50$

Alt Görev 2 (41 Puan)

- $1 \leq N \leq 1000$
- $1 \leq M \leq 5000$

Alt Görev 3 (7 Puan)

- $1 \leq N \leq 100\ 000$
- $M = N - 1$
- Şehrin haritası bir çizgi şeklinde olacaktır. (Tüm şehirler birbirine bağlıdır ve 2 şehir hariç her şehrin iki komşusu vardır.)

Alt Görev 4 (7 Puan)

- $1 \leq N \leq 100\ 000$
- $M = N - 1$
- Şehrin haritası bir ağaç şeklinde olacaktır. (Şehirler arasındaki yollar bir döngü oluşturmaz.)

Alt Görev 5 (28 Puan)

- $1 \leq N \leq 100\ 000$
- $1 \leq M \leq 500\ 000$

Örnek

Örnek Girdi:

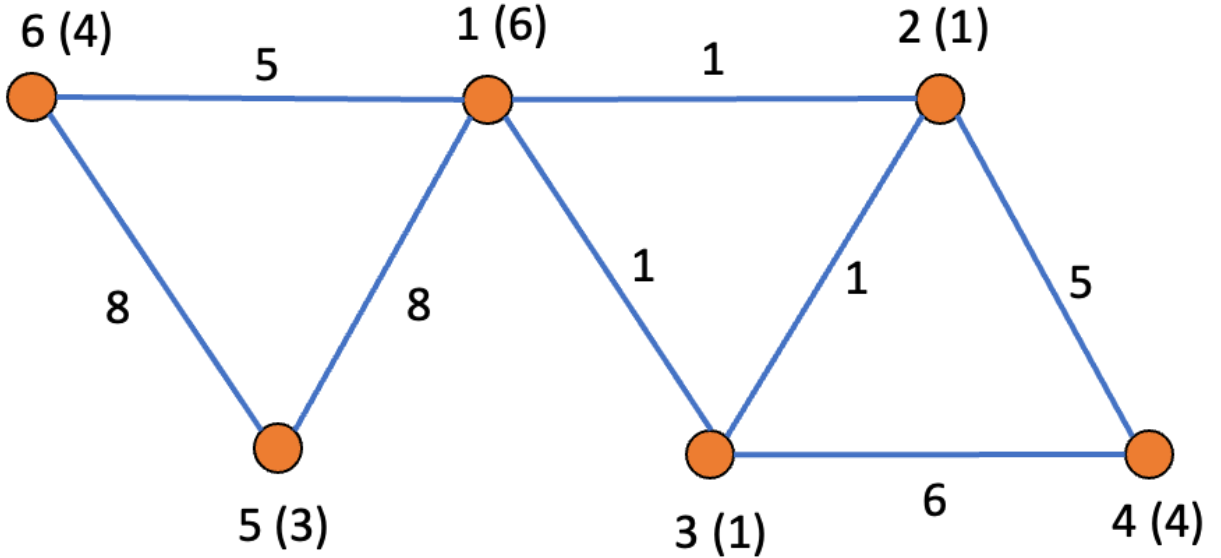
6 8
6 1 1 4 3 4
1 3 1
3 2 1
6 5 8
1 2 1
3 4 6
2 4 5
1 5 8
6 1 5

Örnek Çıktı:

4

Örneğin Açıklaması:

Haritanın görünümü aşağıdaki figürdeki gibidir. Şehirlerin numaralarının yanında parantez içinde ilgili şehrin tahkimatının ne kadar süreceği belirtilmiştir. Ayrıca yolların yanında kaçınıcı saatten itibaren o yolun kullanılmaz hale geleceği belirtilmiştir. Tahkimatlar 2,5,6,1 sırasında yapıldığında 4 şehir için sorunsuz şekilde işlemler gerçekleştirilebilecektir.



Güzel Çizge

Zaman kısıtı: 1.5 sn

Hafıza kısıtı: 256 MB

Size **N** tane düğümden (1'den **N**'e numaralandırılmış) ve **M** tane **ağırlıklı** kenardan oluşan bir çizge veriliyor. Ve bu çizgenin üzerinde gerçekleşecek olan **Q** adet güncelleme var. Sizden istenen her güncellemeden sonra çizgenin **güzellik seviyesini** ekrana yazdırmanız.

Çizgenin **güzellik seviyesi** aşağıdaki şartı sağlayan tüm **pozitif X tam** sayılarının sayısı olarak tanımlanıyor. Şart ise şu şekilde;

Başlangıçta bir düğümde bir jeton olsun. Bu jeton şu anda bulunduğu düğümden ona kenarı olan bir düğüme(kendisi de olabilir) yalnız o kenarın ağırlığı **X**'ten **büyük veya eşitse** taşınabilir ve bu taşıma sonucunda o kenarın ağırlığı **X** kadar **azalır**. Bu şekilde jeton istenilen herhangi bir şekilde taşınabilir. Burada aranan şart ise bir dizi jeton hareketleri sonucunda çizgedeki **tüm kenarların ağırlığının sıfıra** eşit olabilmesidir.

Her güncelleme aşağıdaki iki tipten birinde olabilir:

1. + u v w

2. - u v w

1. tip güncelleme çizgeye **u.** düğüm ve **v.** düğüm arasına **w** ağırlığında bir kenar ekliyor.

2. tip güncelleme çizgeden **u.** düğüm ve **v.** düğüm arasındaki **w** ağırlığında bir kenarı çıkartıyor. Bu işlemden önce çizgede **u.** düğüm ve **v.** düğüm arasında **w** ağırlığında kenar olduğu garanti edilmektedir.

Not: Kenarlar tekrar edebilir ve bir düğümden kendisine kenar olabilir.

Girdi:

Her test birden fazla test durumu içermektedir.

1. satırda bir adet tam sayı **T.** (test durumu sayısı)

Her test durumu;

İlk satırında üç adet tam sayı, sırasıyla **N, M, Q.** (düğüm sayısı, kenar sayısı, güncelleme sayısı)

Takip eden **M** adet satırın her birinde 3 adet tam sayı, sırasıyla **a, b, c.**(çizgede **a.** düğüm ve **b.** düğüm arasında **c** ağırlığında bir kenar olduğunu belirtiyor)

Takip eden **Q** adet satırın her birinde 1 adet '+' ya da '-' ve ardından 3 adet tam sayı, sırasıyla **u, v, w.**(metinde belirtilen güncelleme)

Çıktı:

Her test durumu için:

Q tane tam sayı yazdırın - Her bir güncelleme sonrası çizgenin **güzellik seviyesi**. Eğer **güzellik seviyesi** sonsuzsa -1 yazdırın.

Kısıtlar

$$1 \leq T \leq 100000$$

$$1 \leq N, M, Q \leq 100000$$

$$1 \leq a, b \leq N$$

$$1 \leq c \leq 2000000$$

$$1 \leq u, v \leq N$$

$$1 \leq w \leq 2000000$$

- Tüm N'lerin toplamı ≤ 100000
- Tüm M'lerin toplamı ≤ 100000
- Tüm Q'lerin toplamı ≤ 100000

Alt Görevler

Alt Görev 1 [17 puan]

$$Q = 1, c = 1, w = 1$$

Alt Görev 2 [23 puan]

$$Q = 1$$

Alt Görev 3 [24 puan]

$$c = 1, w = 1$$

Alt Görev 4 [36 puan]

Ek kısıt yoktur.

Örnek Girdi - 1

```
2
1 1 1
1 1 1
- 1 1 1
4 3 5
1 2 4
2 4 4
4 1 4
+ 1 3 8
+ 1 2 4
- 1 4 4
- 1 2 4
- 1 2 4
```

Örnek Çıktı - 1

```
-1
3
3
3
3
0
```

Örnek Açıklama - 1

İlk durumda güzellik seviyesi sonsuz, o yüzden çıktı olarak -1 yazdırılmış.

İkinci durumda ilk 4 güncelleme için X'i 1, 2 ve 4 olarak seçebiliriz, bu sebeple çıktı 3.
5. güncellemede koşulları sağlayan bir X seçmek imkansız olduğu için çıktı 0.

Örnek Girdi - 2

```
1
3 1 5
1 3 4
+ 3 3 4
+ 1 3 2
+ 1 1 2
+ 1 3 1
+ 2 1 5
```

Örnek Çıktı - 2

```
3
2
2
1
1
```