



## Kapanış Saati

Macaristan, 0 ile  $N - 1$  arasında numaralandırılmış  $N$  şehri olan bir ülkedir.

Şehirler, 0 ile  $N - 2$  arasında numaralandırılmış  $N - 1$  adet *çift yönlü* yolla birbirine bağlıdır.  $0 \leq j \leq N - 2$  olacak şekilde her  $j$  için,  $j$  yolu,  $U[j]$  şehri ile  $V[j]$  şehrini birbirine bağlar ve  $W[j]$  uzunluğuna sahiptir. Yani, bir kişinin bu şehirler arasında  $W[j]$  birim zamanda seyahat etmesine olanak tanır. Her yol iki farklı şehri birbirine bağlar ve her şehir çifti en fazla bir yolla birbirine bağlanır.

Birbirinden farklı  $a$  ve  $b$  şehirleri arasındaki **patika**, farklı şehirlerden oluşan  $p_0, p_1, \dots, p_t$  serisidir ve bu seri aşağıdaki şartları sağlar:

- $p_0 = a$ ,
- $p_t = b$ ,
- her  $i$  ( $0 \leq i < t$ ) için  $p_i$  ve  $p_{i+1}$  şehirlerini birbirine bağlayan bir yol vardır.

Herhangi bir şehirden herhangi bir şehre yolları kullanarak ulaşmak mümkündür, yani her iki şehir arasında bir patika bulunmaktadır. Bu patikanın her bir farklı şehir çifti için benzersiz (unique) olduğunu unutmayın.

$p_0, p_1, \dots, p_t$  patikasının **uzunluğu**, patika boyunca ardışık şehirleri birbirine bağlayan  $t$  yolun uzunlukları toplamıdır.

Macaristan'da birçok kişi, iki büyük şehirdeki Kuruluş Günü kutlamalarına katılmak için seyahat eder ve kutlamaları ardından evlerine döner. Kalabalığın halkı rahatsız etmesini engellemek isteyen hükümet, tüm şehirleri belirli saatlerde karantinaya almayı planlıyor. Her şehre hükümet tarafından negatif olmayan bir **kapanış saati** atanacaktır. Hükümet, tüm kapanış saatlerinin toplamının  $K$ 'dan fazla olmaması gerektiğine karar verdi. Başka bir deyişle, 0 ile  $N - 1$  (dahil) arasındaki her  $i$  için,  $i$  şehrine atanan kapanış saati negatif olmayan bir  $c[i]$  tamsayıdır. Tüm  $c[i]$ 'ların toplamı  $K$ 'dan büyük olmamalıdır.

Bir  $a$  şehrini ve bir kapanış saatleri atamasını düşünün. Bir  $b$  şehrinin  $a$  şehrinden **ulaşılabilir** olması demek  $b = a$  olması ya da  $a$  ve  $b$  şehirleri arasında aşağıdaki şartları sağlayan bir  $p_0, \dots, p_t$  patikasının ( $p_0 = a$  ve  $p_t = b$ ) olması demektir.

- $p_0, p_1$  patikasının uzunluğu en fazla  $c[p_1]$ 'dir ve
- $p_0, p_1, p_2$  patikasının uzunluğu en fazla  $c[p_2]$ 'dir ve
- ...

- $p_0, p_1, p_2, \dots, p_t$  patikasının uzunluğu en fazla  $c[p_t]$ 'dir.

Bu yıl, iki ana festival alanı  $X$  şehrinde ve  $Y$  şehrinde bulunuyor. Her kapanış saati ataması için **uygunluk puanı** aşağıdaki iki sayının toplamı olarak tanımlanır:

- $X$  şehrinde ulaşılabilen şehir sayısı.
- $Y$  şehrinde ulaşılabilen şehir sayısı.

Bir şehre hem  $X$  şehrinde hem de  $Y$  şehrinde ulaşılabilirse, bu şehrin uygunluk puanı hesaplamasında *iki kere* sayılacağını unutmayın.

Göreviniz kapanış saati atamalarıyla elde edilebilecek maksimum uygunluk puanını hesaplamaktır.

## Kodlama Detayları

Aşağıdaki prosedürü kodlamalısınız.

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

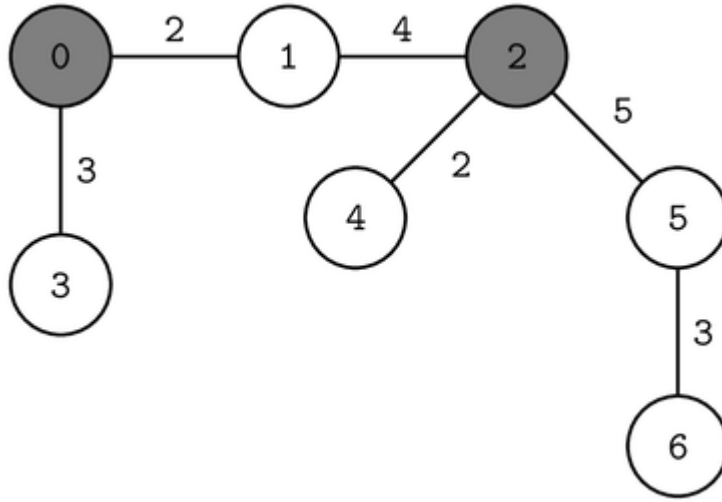
- $N$ : şehir sayısı.
- $X, Y$ : ana festival alanı bulunan şehirler.
- $K$ : kapanış saatleri toplamı üzerindeki üst sınır.
- $U, V$ : yol bağlantılarını gösteren  $N - 1$  uzunluklu diziler.
- $W$ : yol uzunluklarını gösteren  $N - 1$  uzunluklu dizi.
- Bu prosedür bir kapanış saatleri atamasıyla elde edilebilecek en yüksek uygunluk puanını dönmelidir.
- Bazı test durumlarında bu prosedür **birden fazla kere** çağrılabilir.

## Örnek

Aşağıdaki çağrıya bakınız:

```
max_score(7, 0, 2, 10,
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

Bu çağrı aşağıdaki yol ağına karşılık gelmektedir:



Kapanış saati atamalarının aşağıdaki gibi yapıldığını varsayın:

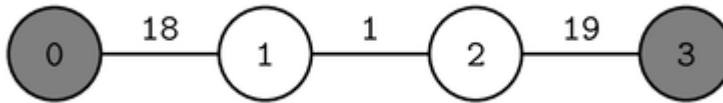
Şehir	0	1	2	3	4	5	6
Kapanış Saati	0	4	0	3	2	0	0

Kapanış saatleri toplamalarının 9 olduğuna dikkat ediniz ve bu toplam  $K = 10$  değerinden büyük değildir. 0, 1 ve 3 numaralı şehirler  $X$  ( $X = 0$ ) şehrinde ulaşılabilir. 1, 2 ve 4 numaralı şehirler ise  $Y$  ( $Y = 2$ ) şehrinde ulaşılabilir. Bu yüzden uygunluk puanı  $3 + 3 = 6$ 'dır. Uygunluk puanı 6'dan büyük olan bir kapanış saati ataması yoktur ve bu yüzden prosedür 6 sayısını dönmelidir.

Ayrıca şu çağrıyı da göz önünde bulundurun:

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

Bu çağrı aşağıdaki yol ağına karşılık gelmektedir:



Kapanış saati atamalarının aşağıdaki gibi yapıldığını varsayın:

Şehir	0	1	2	3
Kapanış Saati	0	1	19	0

0 numaralı şehir  $X$  ( $X = 0$ ) şehrinde ulaşılabilir. 2 ve 3 numaralı şehirler ise  $Y$  ( $Y = 3$ ) şehrinde ulaşılabilir. Bu yüzden uygunluk puanı  $1 + 2 = 3$ 'dür. Uygunluk puanı 3'den büyük

olan bir kapanış saati ataması yoktur ve bu yüzden prosedür 3 sayısını dönmelidir.

## Kısıtlar

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$  ( $0 \leq j \leq N - 2$  şartını sağlayan her  $j$  için)
- $1 \leq W[j] \leq 10^6$  ( $0 \leq j \leq N - 2$  şartını sağlayan her  $j$  için)
- Yolları kullanarak her şehir ikilisi arasında seyahat etmek mümkündür.
- $S_N \leq 200\,000$ ; burada  $S_N$ , bir test durumundaki tüm max\_score çağrılarının  $N$  değerleri toplamıdır.

## Alt görevler

Eğer  $i$  yolu  $i$  ve  $i + 1$  şehirlerini birbirine bağlıyorsa ( $0 \leq i \leq N - 2$  şartını sağlayan her  $i$  için) bir yolu ağının **doğrusal** olduğunu söyleriz.

1. (8 puan)  $X$  şehrinde  $Y$  şehrine olan patikanın uzunluğu  $2K$ 'dan büyüktür.
2. (9 puan)  $S_N \leq 50$ 'dir ve yol ağı doğrusaldır.
3. (12 puan)  $S_N \leq 500$ 'dür ve yol ağı doğrusaldır.
4. (14 puan)  $S_N \leq 3\,000$ 'dir ve yol ağı doğrusaldır.
5. (9 puan)  $S_N \leq 20$
6. (11 puan)  $S_N \leq 100$
7. (10 puan)  $S_N \leq 500$
8. (10 puan)  $S_N \leq 3\,000$
9. (17 puan) Başkaca kısıt yoktur.

## Örnek Değerlendirici

$C$  senaryoların sayısını, yani 'max\_score'a yapılan çağrılarının sayısını gösterebilir. Örnek değerlendirici girdiyi aşağıdaki formatta okur:

- satır 1:  $C$

Daha sonra  $C$  senaryonun tanımlamaları gelir.

Örnek değerlendirici her bir senaryonun tanımlamasını aşağıdaki formatta okur:

- satır 1:  $N\ X\ Y\ K$
- satır 2 +  $j$  ( $0 \leq j \leq N - 2$ ):  $U[j]\ V[j]\ W[j]$

Örnek değerlendirici her senaryo için aşağıdaki formatta tek bir satır yazdırır:

- satır 1: max\_score prosedürünün döndüğü değer