

İdeal şehir

Çağının diğer bilgin ve sanatçıları gibi Leonardo'da şehir plancılığı ve tasarımına ilgi duymaktaydı. Onun amacı ideal bir şehir tasarlamaktı: konforlu, ferah ve kaynak kullanımında akılcı, Ortaçağ'ın dar ve boğucu şehirlerinin çok ötesinde.

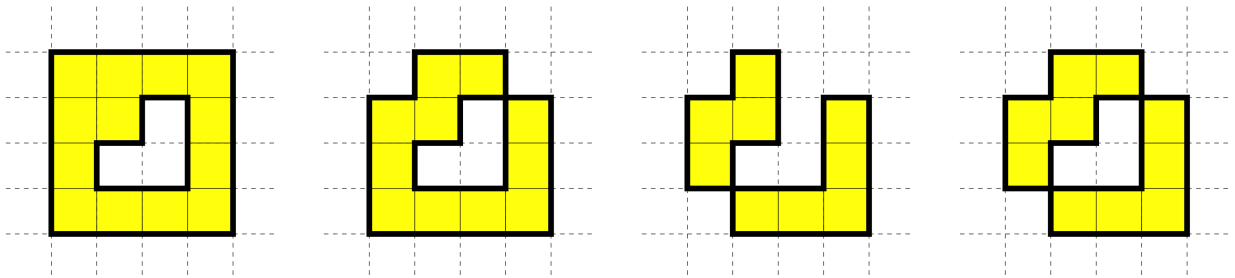
İdeal şehir

Şehir, hayali bir sonsuz ızgara üzerinde N adet bloktan oluşmaktadır. Her bir hücre bir (sıra, sütun) çifti koordinatıyla belirlenir. (i, j) hücresi verildiğinde birbirine komşu şehirler: $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, ve $(i, j + 1)$ dir. Her bir blok ızgara üzerine konulduğunda tam olarak bir hücre kaplar. Bir blok (i, j) hücresine ancak ve ancak $1 \leq i, j \leq 2^{31} - 2$ ise konulabilir. Bir bloğu belirtmek için bloğun üzerinde yer aldığı hücrenin koordinatları kullanılır. İki blok eğer komşu hücrelere yerleştirilmişse komşudur. İdeal bir şehirde, tüm bloklar birbirine kendi sınır blokları arasında “boşluk” olmayacak şekilde bağlıdır, başka bir ifadeyle şu iki koşul sağlanmak zorundadır.

- Herhangi iki *boş* hücre için, bunları birbirine bağlayan en az bir adet komşu *boş* hücreler sıralısı vardır.
- Herhangi iki *boş-olmayan* hücre için, bunları birbirine bağlayan en az bir adet komşu *boş-olmayan* hücreler sıralısı vardır.

Örnek 1

Aşağıdaki blokların hiç birisi bir ideal şehir konfigürasyonu değildir: soldaki ikisi ilk koşulu sağlamaz, üçüncüsü ikinci koşulu sağlamaz, ve dördüncüsü iki koşuluda sağlamaz.

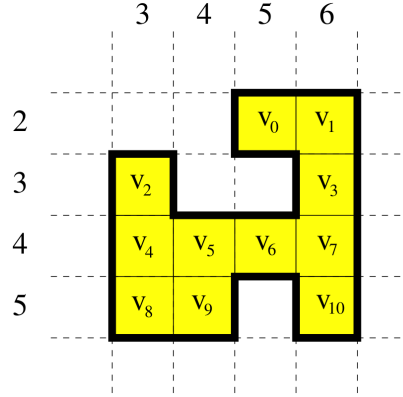


Mesafe

Şehri gezinirken, bir *zıplama* bir bloktan komşu başka bir bloğa gitmeyi gösterir. Boş hücreler gezinilemez. v_0, v_1, \dots, v_{N-1} koordinatları N adet bloğun yerleştirildiği koordinatlar olsun. Birbirinden farklı iki koordinat v_i ve v_j için bunlar arasındaki mesafe $d(v_i, v_j)$, bir bloktan diğerine gitmek için gerekli en az zıplama sayısıdır.

Örnek 2

Aşağıda verilen konfigürasyon $N = 11$ bloktan oluşan ve blokların $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, ve $v_{10} = (5, 6)$ koordinatlarında yer aldığı bir ideal şehirdir. Örneğin, $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$, ve $d(v_9, v_{10}) = 4$.



Görev ifadesi

Sizin göreviniz, ideal bir şehir verildiğinde, $i < j$ olacak şekilde tüm blok çiftleri (v_i and v_j) arasındaki mesafelerin toplamını bulmaktır. Yani, aşağıda verilen toplamın değerini hesaplamalısınız:

$$\sum d(v_i, v_j), \text{ burada } 0 \leq i < j \leq N - 1$$

N değeri, ve X ve Y dizileri verildiğinde, yazmanız gereken fonksiyon $\text{DistanceSum}(N, X, Y)$ yukarıdaki formülü hesaplamalıdır. X ve Y dizilerinin her ikisi de N elemanlıdır: blok i , $(X[i], Y[i])$ koordinatında bulunur öyleki $0 \leq i \leq N - 1$, ve $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Sonuç çok büyük olabileceğinden 32 bit ile gösterilemeyebilir, dolayısıyla sonucun 1 000 000 000 (bir milyar) a göre modunu döndürmelisiniz.

Örnekte, $11 \times 10 / 2 = 55$ blok çifti vardır. Blok çiftleri arası ikili mesafelerin toplamı 174 dür.

Altgörev 1 [11 puan]

$N \leq 200$ kabul edebilirsiniz.

Altgörev 2 [21 puan]

$N \leq 2\,000$ kabul edebilirsiniz.

Altgörev 3 [23 puan]

$N \leq 100\,000$ kabul edebilirsiniz.

İlave olarak verilen şu iki koşul sağlanır: $X[i] = X[j]$ olacak şekilde i ve j boş-olmayan herhangi iki hücre ise bunların arasındaki tüm hücreler de boş değildir; $Y[i] = Y[j]$ olacak şekilde i ve j boş-olmayan herhangi iki hücre ise bunların arasındaki tüm hücreler de boş değildir

Altgörev 4 [45 puan]

$N \leq 100\,000$ kabul edebilirsiniz.

Gerçekleştirim detayları

Adı `city.c`, `city.cpp` veya `city.pas` olan yalnız bir dosya göndermelisiniz. Bu dosya yukarıda belirtilen fonksiyonu aşağıda verilen başlık yapısında gerçekleştirmelidir.

C/C++ programı

```
int DistanceSum(int N, int *X, int *Y);
```

Pascal programı

```
function DistanceSum(N : LongInt; var X, Y : array of LongInt) : LongInt;
```

Belirtilen fonksiyon yukarıda anlatıldığı gibi çalışmalıdır. Bunlar haricinde kendiniz ilave yardımcı fonksiyonlar da yazabilirsiniz. Kodunuz hiç bir şekilde standart girdi/çıkı ve başka bir dosyadan okuma yazma yapmamalıdır.

Örnek notlandırıcı

Örnek notlandırıcı girdiyi aşağıdaki formatta okumaktadır.

- satır 1: N ;
- satırlar 2, ..., $N + 1$: $X[i]$, $Y[i]$.

Süre ve hafıza limiti

- Süre limiti: 1 saniye.
- Hafıza limiti: 256 MB.