

Nil

N tane eseri Nil nehri kullanarak taşımak istiyorsunuz. Eserler 0'dan $N - 1$ 'e kadar numaralandırılmıştır. $0 \leq i < N$ olmak üzere i numaralı eserin ağırlığı $W[i]$ 'dir.

Eserleri taşımak için özel botlar kullanıyorsunuz ve her both **en fazla iki** adet eser taşıyabilir.

- Bir bota tek bir eser koymak istiyorsanız, eserin ağırlığı herhangi bir sayı olabilir.
- Eğer iki eseri aynı bota koymak isterseniz, botun dengeli olmasını sağlamanız gerekmektedir. Daha açık bir ifadeyle, $0 \leq p < q < N$ olmak üzere eser p ve q 'nun aynı bota koyulabilmeleri için ağırlıklarının mutlak farkının en fazla D olması ($|W[p] - W[q]| \leq D$) gerekmektedir.

Bir eseri taşımanız için ödemeniz gereken bedel o eserle beraber aynı botta başka esrin taşınıp taşınmamasına bağlıdır.

$0 \leq i < N$ olmak üzere eser i 'yi taşınanın bedeli:

- eser i tek başına taşınıyorsa $A[i]$,
- eser i başka bir eserle beraber taşınıyorsa $B[i]$ 'dir.

İkinci durumda bottaki her iki eser için de ödeme yapmanız gerektiğine dikkat ediniz. Daha açık bir ifadeyle, eser p ve q 'yu aynı botla taşırsanız, yapmanız gereken ödeme $B[p] + B[q]$ 'dur.

Bir eseri tek başına bir botla taşımak her zaman başka bir eserle beraber taşımaktan daha pahalıdır. Yani, $0 \leq i < N$ olmak üzere her i için $B[i] < A[i]$ 'dir.

Ne yazık ki, nehir öngörülebilir değildir ve D değeri sıklıkla değişmektedir. Sizin göreviniz 0 ve $Q - 1$ arasında numaralandırılmış Q adet soruyu cevaplamaktır. Sorular Q uzunluklu bir E dizisiyle tanımlanmıştır. $0 \leq j < Q$ olmak üzere j numaralı sorunun cevabı D 'nin $E[j]$ 'ye eşit olduğu durumda N adet eserin tamamını taşımak için yapılması gereken en düşük toplam ödeme miktarıdır.

Uygulama Detayları

Aşağıdaki prosedürü uygulamalısınız.

```
std::vector<long long> calculate_costs(  
    std::vector<int> W, std::vector<int> A,  
    std::vector<int> B, std::vector<int> E)
```

- W, A, B : Bunlar N uzunluklu diziler olup eserlerin ağırlıklarını ve taşıma bedellerini göstermektedirler.
- E : Q uzunluklu bir dizi olup her soru için D değerini vermektedir.
- Bu prosedür Q uzunluklu R dizisini döndürmelidir. $0 \leq j < Q$ olmak üzere $R[j]$ 'nin değeri D 'nin $E[j]$ 'ye eşit olduğu durumda N adet eser taşımak için yapılması gereken en düşük toplam ödeme miktarı olmalıdır.
- Bu prosedür her test durumu için tam olarak bir defa çalıştırılacaktır.

Kısıtlar

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq i < N$ olmak üzere her i için $1 \leq W[i] \leq 10^9$ 'dur.
- $0 \leq i < N$ olmak üzere her i için $1 \leq B[i] < A[i] \leq 10^9$ 'dur.
- $0 \leq j < Q$ olmak üzere her j için $1 \leq E[j] \leq 10^9$ 'dir.

Alt görevler

Alt görev	Skor	Ek Kısıtlar
1	6	$Q \leq 5; N \leq 2000; 0 \leq i < N$ olmak üzere her i için $W[i] = 1$
2	13	$Q \leq 5; 0 \leq i < N$ olmak üzere her i için $W[i] = i + 1$
3	17	$Q \leq 5; 0 \leq i < N$ olmak üzere her i için $A[i] = 2$ ve $B[i] = 1$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$0 \leq i < N$ olmak üzere her i için $A[i] = 2$ ve $B[i] = 1$
7	18	Başkaca kısıt yoktur.

Örnek

Aşağıdaki çağrıyı inceleyelim.

```
calculate_costs([15, 12, 2, 10, 21],
               [5, 4, 5, 6, 3],
               [1, 2, 2, 3, 2],
               [5, 9, 1])
```

Bu örnekte $N = 5$ eser ve $Q = 3$ soru vardır.

İlk soruda $D = 5$ 'dir. Eser 0 ve 3'ü bir botla (çünkü $|15 - 10| \leq 5$ 'tir) ve geri kalan eserleri ise ayrı botlarla taşıyabilirsiniz. Bu şekilde bütün eserleri en düşük toplam maliyetle

$(1 + 4 + 5 + 3 + 3 = 16)$ taşıyabilirsiniz.

İkinci soruda $D = 9$ 'dur. Eser 0 ve 1'ü bir botla (çünkü $|15 - 12| \leq 9$ 'dur) ve eser 2 ve 3'ü ise başka bir botla (çünkü $|2 - 10| \leq 9$ 'dur) taşıyabilirsiniz. Kalan eser ise ayrı bir botla taşınabilir. Bu şekilde bütün eserleri en düşük toplam maliyetle $(1 + 2 + 2 + 3 + 3 = 11)$ taşıyabilirsiniz.

Son soruda $D = 1$ 'dir. Her eseri ayrı bir botla taşımanız gerekmektedir. Bu şekilde bütün eserleri en düşük toplam maliyetle $(5 + 4 + 5 + 6 + 3 = 23)$ taşıyabilirsiniz.

Bu yüzden, bu prosedür $[16, 11, 23]$ dizisini dönmelidir.

Örnek Değerlendirici

Girdi formatı:

```
N
W[0] A[0] B[0]
W[1] A[1] B[1]
...
W[N-1] A[N-1] B[N-1]
Q
E[0]
E[1]
...
E[Q-1]
```

Çıktı formatı:

```
R[0]
R[1]
...
R[S-1]
```

Burada S `calculate_costs` tarafından dönen R dizisinin uzunluğudur.