

ナイル川 (Nile)

あなたは N 個の遺物をナイル川の流にらせて運送しようとしている。遺物には 0 から $N - 1$ までの番号が付けられている。遺物 i ($0 \leq i < N$) の重さは $W[i]$ である。

あなたはたくさんの専用のボートを用いて遺物を運送する。それぞれのボートは **最大 2 つ** の遺物を運ぶことができる。

- 1 つの遺物を単独でボートに載せる場合、その遺物の重さは何であっても良い。
- 2 つの遺物を同じボートに載せたい場合、あなたはそのボートのバランスが保たれるようにしなければならない。具体的には、2 つの遺物 p, q ($0 \leq p < q < N$) は、それらの重さの差の絶対値が D 以下であるとき、すなわち $|W[p] - W[q]| \leq D$ であるときに限り、同じボートで運送することができる。

ある遺物を運送するには、その遺物と同じボートで運ばれる遺物の数に応じたコストがかかる。遺物 i ($0 \leq i < N$) を運送するコストは、

- その遺物を単独でボートに載せる場合、 $A[i]$ である。
- その遺物を他の遺物と同じボートに載せる場合、 $B[i]$ である。

なお、後者の場合においては、同じボートに載せる 2 つの遺物それぞれに対してコストがかかる。具体的には、2 つの遺物 p, q ($0 \leq p < q < N$) を同じボートで運送する場合、 $B[p] + B[q]$ のコストがかかる。

ある遺物を単独でボートに載せる場合にかかるコストは、他の遺物と同じボートに載せる場合のコストより常に大きい。すなわち、すべての i ($0 \leq i < N$) について、 $B[i] < A[i]$ が成り立つ。

残念ながら、ナイル川はとても気まぐれな川なので、 D の値はしばしば変化する。あなたの課題は、 0 から $Q - 1$ までの番号が付けられた Q 個の質問に答えることである。これらの質問は長さ Q の配列 E によって表される。質問 j ($0 \leq j < Q$) に対する答えは、 D の値が $E[j]$ に等しいときの、 N 個の遺物すべてを運送するのにかかるコストの合計の最小値である。

実装の詳細

あなたは以下の関数を実装する必要がある。

```
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 : それぞれが長さ N の整数列であり, 各遺物の重さおよびそれらを運送するのにかかるコストを表す.
- E : 長さ Q の整数列であり, それぞれの質問における D の値を表す.
- この関数は長さ Q の整数列 R を返す必要がある. ここで, それぞれの j ($0 \leq j < Q$) について, $R[j]$ は D の値が $E[j]$ であるときにすべての遺物を運送するのにかかるコストの合計の最小値を表す.
- この関数は各テストケースにおいてちょうど 1 回呼び出される.

制約

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ ($0 \leq i < N$)
- $1 \leq B[i] < A[i] \leq 10^9$ ($0 \leq i < N$)
- $1 \leq E[j] \leq 10^9$ ($0 \leq j < Q$)

小課題

小課題	得点	追加の制約
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ ($0 \leq i < N$)
2	13	$Q \leq 5; W[i] = i + 1$ ($0 \leq i < N$)
3	17	$Q \leq 5; A[i] = 2$ かつ $B[i] = 1$ ($0 \leq i < N$)
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ かつ $B[i] = 1$ ($0 \leq i < N$)
7	18	追加の制約はない.

例

以下の呼び出しを考える.

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

この例では $N = 5$ 個の遺物と $Q = 3$ 個の質問がある.

質問 0 では, $D = 5$ である. あなたは遺物 0, 3 を同じボートで運び ($|15 - 10| \leq 5$ のため可能である), 残りの遺物を別々のボートで運ぶことができる. この際にかかるコストは

$1 + 4 + 5 + 3 + 3 = 16$ であり、これがすべての遺物を運送するのにかかるコストの合計の最小値である。

質問 1 では、 $D = 9$ である。あなたは遺物 0, 1 を同じボートで運び ($|15 - 12| \leq 9$ のため可能である)、遺物 2, 3 を同じボートで運び ($|2 - 10| \leq 9$ のため可能である)、残りの遺物を単独でボートに載せて運ぶことができる。この際にかかるコストは $1 + 2 + 2 + 3 + 3 = 11$ であり、これがすべての遺物を運送するのにかかるコストの合計の最小値である。

質問 2 では、 $D = 1$ である。あなたはそれぞれの遺物を単独でボートに載せて運ぶ必要がある。この際にかかるコストは $5 + 4 + 5 + 6 + 3 = 23$ であり、これがすべての遺物を運送するのにかかるコストの合計の最小値である。

したがって、この関数は $[16, 11, 23]$ を返す必要がある。

採点プログラムのサンプル

入力形式：

```
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]
```

出力形式：

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

ここで、 S は `calculate_costs` によって返された配列 R の長さである。