Νείλος

Θέλεις να μεταφέρεις N αρχαία αντικείμενα μέσω του Νείλου. Τα αντικείμενα είναι αριθμημένα από 0 έως $N-1$. Το βάρος του αντικειμένου i ($0 \leq i < N$) είναι $W[i]$.

Για να μεταφέρεις τα αντικείμενα, χρησιμοποιείς ειδικά σκάφη. Κάθε σκάφος μπορεί να μεταφέρει **το πολύ δύο** αντικείμενα.

- Εάν αποφασίσεις να τοποθετήσεις ένα μόνο αντικείμενο σε ένα σκάφος, το βάρος του αντικειμένου μπορεί να είναι αυθαίρετο.
- Εάν θέλεις να τοποθετήσεις δύο αντικείμενα στο ίδιο σκάφος, πρέπει να διασφαλίσεις ότι το σκάφος είναι ισοζυγισμένο. Συγκεκριμένα, μπορείς να στείλεις τα αντικείμενα p και q ($0 \leq p < q < N$) στο ίδιο σκάφος μόνο εάν η απόλυτη διαφορά μεταξύ των βαρών τους είναι το πολύ D, δηλαδή $|W[p] - W[q]| \leq D$.

Για να μεταφέρεις ένα αντικείμενο, πρέπει να πληρώσεις ένα κόστος που εξαρτάται από τον αριθμό των αντικειμένων που μεταφέρονται στο ίδιο σκάφος. Το κόστος μεταφοράς του αντικειμένου i ($0 \leq i < N$) είναι:

- $A[i]$, εάν τοποθετήσεις το αντικείμενο μόνο του σε ένα σκάφος, ή
- $B[i]$, εάν το τοποθετήσεις σε σκάφος μαζί με κάποιο άλλο αντικείμενο.

Σημείωσε ότι στην τελευταία περίπτωση, πρέπει να πληρώσεις για τα δύο αντικείμενα στο σκάφος. Συγκεκριμένα, εάν αποφασίσεις να στείλεις τα αντικείμενα p και q ($0 \leq p < q < N$) στο ίδιο σκάφος, πρέπει να πληρώσεις $B[p] + B[q]$.

Η αποστολή ενός αντικειμένου σε σκάφος μόνο του είναι πάντα πιο ακριβή από το να το στείλεις με κάποιο άλλο αντικείμενο που μοιράζεται το σκάφος, οπότε $B[i] < A[i]$ για όλα τα i τέτοια ώστε $0 \leq i < N$.

Δυστυχώς, ο ποταμός είναι πολύ απρόβλεπτος και η τιμή του D αλλάζει συχνά. Η αποστολή σου είναι να απαντήσεις σε Q ερωτήσεις αριθμημένες από 0 έως $Q-1$. Οι ερωτήσεις περιγράφονται από έναν πίνακα E μήκους Q. Η απάντηση στην ερώτηση j ($0 \leq j < Q$) είναι το ελάχιστο συνολικό κόστος μεταφοράς όλων των N αντικειμένων, όταν η τιμή του D είναι ίση με $E[j]$.

Λεπτομέρειες Υλοποίησης

Πρέπει να υλοποιήσεις την εξής διαδικασία.

```
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: πίνακες ακεραίων μήκους N, που περιγράφουν τα βάρη των αντικειμένων και τα κόστη μεταφοράς τους.
- E: ένας πίνακας ακεραίων μήκους Q που περιγράφει την τιμή του D για κάθε ερώτηση.
- Αυτή η διαδικασία πρέπει να επιστρέψει έναν πίνακα R με Q ακέραιους που περιέχει το ελάχιστο συνολικό κόστος μεταφοράς των αντικειμένων, όπου το $R[j]$ δίνει το κόστος όταν η τιμή του D είναι $E[j]$ (για κάθε j τέτοια ώστε $0 \leq j < Q$).
- Αυτή η διαδικασία καλείται ακριβώς μία φορά για κάθε testcase.

Περιορισμοί

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ για κάθε i τέτοιο ώστε $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ για κάθε i τέτοιο ώστε $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ για κάθε j τέτοιο ώστε $0 \leq j < Q$

Υποερωτήματα

Υποερώτημα	Βαθμολογία	Πρόσθετοι Περιορισμοί
1	6	$Q \leq 5$; $N \leq 2000$; $W[i] = 1$ για κάθε i τέτοιο ώστε $0 \leq i < N$
2	13	$Q \leq 5$; $W[i] = i + 1$ για κάθε i τέτοιο ώστε $0 \leq i < N$
3	17	$Q \leq 5$; $A[i] = 2$ και $B[i] = 1$ για κάθε i τέτοιο ώστε $0 \leq i < N$
4	11	$Q \leq 5$; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ και $B[i] = 1$ για κάθε i τέτοιο ώστε $0 \leq i < N$
7	18	Χωρίς πρόσθετους περιορισμούς.

Παράδειγμα

Ας εξετάσουμε την εξής κλήση.

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

Σε αυτό το παράδειγμα έχουμε $N = 5$ αντικείμενα και $Q = 3$ ερωτήσεις.

Στην πρώτη ερώτηση, $D = 5$. Μπορείς να στείλεις τα αντικείμενα 0 και 3 σε ένα σκάφος (καθώς $|15 - 10| \leq 5$) και τα υπόλοιπα αντικείμενα σε ξεχωριστά σκάφη. Αυτό δίνει το ελάχιστο κόστος μεταφοράς όλων των αντικειμένων, που είναι $1 + 4 + 5 + 3 + 3 = 16$.

Στη δεύτερη ερώτηση, $D = 9$. Μπορείς να στείλεις τα αντικείμενα 0 και 1 σε ένα σκάφος (καθώς $|15 - 12| \leq 9$) και να στείλεις τα αντικείμενα 2 και 3 σε ένα σκάφος (καθώς $|2 - 10| \leq 9$). Το τελευταίο αντικείμενο μπορεί να σταλεί σε ξεχωριστό σκάφος. Αυτό δίνει το ελάχιστο κόστος μεταφοράς όλων των αντικειμένων, που είναι $1 + 2 + 2 + 3 + 3 = 11$.

Στην τελευταία ερώτηση, $D = 1$. Πρέπει να στείλεις κάθε αντικείμενο στο δικό του σκάφος. Αυτό δίνει το ελάχιστο κόστος μεταφοράς όλων των αντικειμένων, που είναι $5 + 4 + 5 + 6 + 3 = 23$.

Επομένως, αυτή η διαδικασία πρέπει να επιστρέψει $[16, 11, 23]$.

Υπόδειγμα Grader

Μορφή εισόδου:

```
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]
```

Μορφή εξόδου:

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

Εδώ, S είναι το μήκος του πίνακα R που επιστρέφεται από το `calculate_costs`.