

Nil

Vous désirez transporter N reliques sur le Nil. Les reliques sont numérotées de 0 à $N - 1$. Le poids de la relique i ($0 \leq i < N$) est de $W[i]$.

Pour transporter les reliques, vous devez utiliser des bateaux spéciaux. Chaque bateau peut porter **au plus deux** reliques.

- Si vous décidez de mettre une seule relique dans un bateau, il n'y a pas de restriction sur le poids de cette relique.
- Si vous décidez de mettre deux reliques dans un même bateau, il faut que le bateau soit stable. Plus précisément, vous pouvez charger les reliques p et q ($0 \leq p < q < N$) sur le même bateau seulement si la valeur absolue de la différence entre leur poids est d'au plus D , c'est-à-dire, si $|W[p] - W[q]| \leq D$.

Pour transporter une relique, il faut payer un prix qui dépend du nombre de reliques dans le même bateau. Le transport de la relique i ($0 \leq i < N$) coûte:

Notez que dans le second cas, vous devez payer pour les deux reliques dans le bateau. Plus précisément, si vous décidez d'envoyer les reliques p et q ($0 \leq p < q < N$) dans le même bateau, vous devez payer $B[p] + B[q]$.

Envoyer une relique toute seule dans un bateau est toujours plus cher que de l'envoyer avec une autre relique dans le même bateau, c'est-à-dire, $B[i] < A[i]$ pour tout i tel que $0 \leq i < N$.

Malheureusement, la rivière est très imprévisible et la valeur de D change souvent. Votre tâche est de répondre à Q questions numérotées de 0 à $Q - 1$. Les questions sont décrites dans un tableau E de taille Q . La réponse à la question j ($0 \leq j < Q$) est le coût total minimal du transport des N reliques, lorsque la valeur de D est égale à $E[j]$.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
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 : des tableaux d'entiers de taille N , qui représentent les poids des reliques et les coûts de leur transport.
- E : un tableau d'entiers de taille Q qui représente la valeur de D pour chaque question.
- Cette fonction doit renvoyer un tableau R de Q entiers, qui contient les coûts totaux minimaux des transports des reliques, où $R[j]$ est le coût lorsque la valeur de D est $E[j]$ (pour tout j tel que $0 \leq j < Q$).
- Cette fonction est appelée exactement une fois par test.

Constraints

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ pour tout i tel que $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ pour tout i tel que $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ pour tout j tel que $0 \leq j < Q$

Sous-tâches

Sous-tâche	Score	Contraintes supplémentaires
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ pour tout i tel que $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ pour tout i tel que $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ et $B[i] = 1$ pour tout i tel que $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ et $B[i] = 1$ pour tout i tel que $0 \leq i < N$
7	18	Aucune contrainte supplémentaire.

Exemple

Considérez l'appel de fonction suivant.

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

Dans cet exemple, il y a $N = 5$ reliques et $Q = 3$ questions.

Dans la première question, $D = 5$. Vous pouvez envoyer les reliques 0 et 3 dans un même bateau (puisque $|15 - 10| \leq 5$) et les reliques restantes dans des bateaux séparés. Cela donne un coût

minimal de transport de toutes les reliques de $1 + 4 + 5 + 3 + 3 = 16$.

Dans la deuxième question, $D = 9$. Vous pouvez envoyer les reliques 0 et 1 dans un même bateau (car $|15 - 12| \leq 9$) et envoyer les reliques 2 et 3 dans un même bateau (car $|2 - 10| \leq 9$). La relique restante peut être envoyée dans un bateau séparé. Cela donne un coût minimal de transport de toutes les reliques de $1 + 2 + 2 + 3 + 3 = 11$.

Dans la dernière question, $D = 1$. Vous devez envoyer chaque relique dans son propre bateau. Cela donne un coût minimal de transport de toutes les reliques de $5 + 4 + 5 + 6 + 3 = 23$.

Par conséquent, cet appel de fonction devrait renvoyer $[16, 11, 23]$.

Évaluateur d'exemple (grader)

Format d'entrée :

```
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]
```

Format de sortie :

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

Ici, S est la taille du tableau R renvoyé par `calculate_costs`.