



# Réunions (Meetings)

Il y a  $N$  montagnes alignées sur une ligne horizontale, et numérotées de 0 à  $N - 1$  de gauche à droite. La hauteur de la montagne  $i$  est  $H_i$  ( $0 \leq i \leq N - 1$ ). Exactement une personne habite au sommet de chaque montagne.

Vous serez amené à organiser  $Q$  réunions, numérotées de 0 à  $Q - 1$ . Les participants à la réunion  $j$  ( $0 \leq j \leq Q - 1$ ) seront toutes les personnes vivant au sommet des montagnes de  $L_j$  à  $R_j$  inclusivement ( $0 \leq L_j \leq R_j \leq N - 1$ ). Pour cette réunion  $j$ , vous devez choisir une montagne  $x$  comme lieu de réunion ( $L_j \leq x \leq R_j$ ). Cette réunion a un coût, dépendant de votre sélection, qui est calculé comme suit :

- Le coût pour le participant venant de la montagne  $y$  ( $L_j \leq y \leq R_j$ ) est la hauteur maximale de toutes les montagnes entre  $x$  et  $y$ , inclusivement. En particulier, le coût du participant venant de la montagne  $x$  est  $H_x$ , la hauteur de la montagne  $x$ .
- Le coût d'une réunion est la somme des coûts de ses participants.

Pour chaque réunion, on souhaite trouver le coût minimal nécessaire à son organisation.

Notez que tous les participants reviennent à leurs montagnes après chaque réunion ; le coût d'une réunion n'est donc pas influencé par la réunion précédente.

## Détails de l'implémentation

Vous devez définir la fonction suivante:

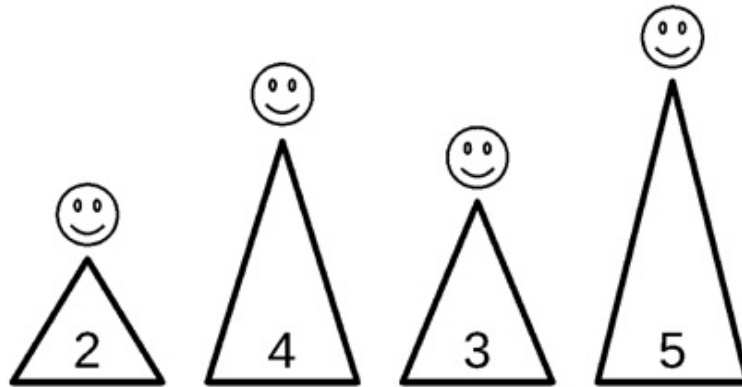
```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

- $H$  : un tableau de taille  $N$ , représentant les hauteurs des montagnes.
- $L$  et  $R$  : des tableaux de longueur  $Q$ , représentant la plage des participants dans les réunions.
- Cette fonction doit retourner un tableau  $C$  de longueur  $Q$ . La case  $C_j$  ( $0 \leq j \leq Q - 1$ ) doit être le coût minimum d'organisation de la réunion  $j$ .
- Notez que les valeurs de  $N$  et  $Q$  sont les longueurs des tableaux et peuvent être obtenues comme indiqué dans la notice d'implémentation.

## Exemple

Soit  $N = 4$ ,  $H = [2, 4, 3, 5]$ ,  $Q = 2$ ,  $L = [0, 1]$ , et  $R = [2, 3]$ .

L'évaluateur appelle `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])`.



Pour la réunion  $j = 0$ , on a  $L_j = 0$  et  $R_j = 2$ , les personnes vivant aux sommets des montagnes 0, 1 et 2 vont y assister. Si la montagne 0 est choisie comme lieu de réunion, le coût de la réunion 0 est calculé comme suit :

- Le coût du participant venant de la montagne 0 est  $\max\{H_0\} = 2$ .
- Le coût du participant venant de la montagne 1 est  $\max\{H_0, H_1\} = 4$ .
- Le coût du participant venant de la montagne 2 est  $\max\{H_0, H_1, H_2\} = 4$ .
- Par conséquent, le coût de la réunion 0 est  $2 + 4 + 4 = 10$

Il est impossible de tenir la réunion 0 à un coût inférieur, le coût minimum de la réunion 0 est donc 10.

Pour la réunion  $j = 1$ , on a  $L_j = 1$  et  $R_j = 3$ , les personnes vivant aux sommets des montagnes 1, 2, et 3 vont y assister. Si la montagne 2 est choisie comme lieu de réunion, le coût de la réunion 1 est calculé comme suit:

- Le coût du participant venant de la montagne 1 est  $\max\{H_1, H_2\} = 4$ .
- Le coût du participant venant de la montagne 2 est  $\max\{H_2\} = 3$ .
- Le coût du participant venant de la montagne 3 est  $\max\{H_2, H_3\} = 5$ .
- Par conséquent, le coût de la réunion 1 est  $4 + 3 + 5 = 12$ .

Il est impossible de tenir la réunion 1 à un coût inférieur, le coût minimum de la réunion 1 est donc 12.

Les fichiers `sample-01-in.txt` et `sample-01-out.txt` dans le package attaché correspondent à cet exemple. D'autres exemples d'entrées / sorties sont également disponibles dans le package.

## Contraintes

- $1 \leq N \leq 750\,000$
- $1 \leq Q \leq 750\,000$

- $1 \leq H_i \leq 1\,000\,000\,000$  ( $0 \leq i \leq N - 1$ )
- $0 \leq L_j \leq R_j \leq N - 1$  ( $0 \leq j \leq Q - 1$ )
- $(L_j, R_j) \neq (L_k, R_k)$  ( $0 \leq j < k \leq Q - 1$ )

## Sous-tâches

1. (4 points)  $N \leq 3\,000$ ,  $Q \leq 10$
2. (15 points)  $N \leq 5\,000$ ,  $Q \leq 5\,000$
3. (17 points)  $N \leq 100\,000$ ,  $Q \leq 100\,000$ ,  $H_i \leq 2$  ( $0 \leq i \leq N - 1$ )
4. (24 points)  $N \leq 100\,000$ ,  $Q \leq 100\,000$ ,  $H_i \leq 20$  ( $0 \leq i \leq N - 1$ )
5. (40 points) pas de contraintes additionnelles

## Évaluateur d'exemple (Sample grader)

L'évaluateur lit l'entrée au format suivant :

- ligne 1:  $N$   $Q$
- ligne 2:  $H_0$   $H_1$   $\cdots$   $H_{N-1}$
- ligne  $3 + j$  ( $0 \leq j \leq Q - 1$ ):  $L_j$   $R_j$

L'évaluateur affiche la valeur de retour de `minimum_costs` au format suivant :

- ligne  $1 + j$  ( $0 \leq j \leq Q - 1$ ):  $C_j$