

Cohésion d'équipe (Team Building)

Votre objectif est de recruter une équipe de N programmeurs. Vous avez déjà repéré et évalué que le niveau de compétence du i -ème individu ($1 \leq i \leq N$) est égal à l'entier positif (ou nul) $s[i]$. Vous avez remarqué que l'ordre des recrutements est important.

Chaque programmeur est caractérisé par deux valeurs supplémentaires : son efficacité et sa motivation, toutes deux égales à 0 lors du recrutement, et qui peuvent augmenter avec l'arrivée de nouveaux programmeurs. Quand un nouveau programmeur est recruté, les événements suivants se produisent dans l'ordre suivant :

- Le nouveau programmeur rejoint l'équipe avec une efficacité et une motivation initialisées à 0.
- L'efficacité de chaque programmeur déjà recruté est augmentée de la valeur de sa propre motivation.
- La motivation de chaque programmeur déjà recruté est augmentée du niveau de compétence de la nouvelle recrue.

L'efficacité de l'équipe est déterminée à la fin, et est égale à la somme des efficacités des membres de l'équipe. Votre objectif est de calculer l'efficacité maximale de l'équipe en optimisant l'ordre des recrutements.

Par exemple, si vous recrutez des programmeurs avec des niveaux de compétence $(0, 2, 2, 3)$, dans cet ordre, le processus de recrutement se déroule comme suit :

Évènement	Efficacités	Motivations
Recrutement niveau 0	0	0
Recrutement niveau 2	0 0	0 0
Modification des efficacités	0 0	0 0
Modification des motivations	0 0	2 0
Recrutement niveau 2	0 0 0	2 0 0
Modification des efficacités	2 0 0	2 0 0
Modification des motivations	2 0 0	4 2 0
Recrutement niveau 3	2 0 0 0	4 2 0 0
Modification des efficacités	6 2 0 0	4 2 0 0
Modification des motivations	6 2 0 0	7 5 3 0

L'efficacité de l'équipe est alors égale à $6 + 2 + 0 + 0 = 8$. Cependant, si vous recrutez dans l'ordre $(2, 2, 3, 0)$, vous obtenez une efficacité d'équipe de $7 + 3 + 0 + 0 = 10$.

Niveau recrue	Efficacités	Motivations
2	0	0
2	0 0	2 0
3	2 0 0	5 3 0
0	7 3 0 0	5 3 0 0

De plus, au cours des Q prochains jours, vous recevrez des modifications de l'évaluation des compétences de certains programmeurs. Après le i -ème jour, l'évaluation du niveau du programmeur $x[i]$ deviendra $y[i]$ (possiblement égal à la valeur précédente). Cette nouvelle évaluation de compétence sera utilisée les jours suivants, jusqu'à ce qu'elle soit possiblement modifiée à nouveau.

Après chaque jour, à partir d'aujourd'hui, votre objectif est de calculer l'efficacité maximale réalisable de l'équipe pour le recrutement des N programmeurs, en prenant en compte l'évaluation des niveaux de compétence à cet instant.

Format de l'entrée

La première ligne contient deux entiers : N et Q .

La deuxième ligne contient les entiers $s[1], s[2], \dots, s[N]$.

Les Q lignes suivantes contiennent chacune deux entiers : $x[i]$ et $y[i]$.

Format de la sortie

Affichez $Q + 1$ lignes, chacune contenant un unique entier. Ces entiers représentent l'efficacité maximale de l'équipe après chaque jour, dans l'ordre chronologique.

Exemple

Entrée standard	Sortie standard
4 2	10
2 0 2 3	14
2 4	12
4 0	

La solution pour l'état initial est représentée ci-dessus. Après le premier jour, l'évaluation des niveaux de compétence devient $(2, 4, 2, 3)$ et l'efficacité d'équipe maximale est alors 14. Après le second jour, l'évaluation des niveaux de compétence devient $(2, 4, 2, 0)$.

Contraintes

- $2 \leq N \leq 50\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq s[i] \leq 100\,000$ pour chaque $1 \leq i \leq N$.
- $1 \leq x[i] \leq N$ pour chaque $1 \leq i \leq Q$.
- $0 \leq y[i] \leq 100\,000$ pour chaque $1 \leq i \leq Q$.

Sous-tâches

1. (11 points) $N \leq 7; Q \leq 100$
2. (19 points) $N, Q \leq 500$
3. (15 points) $Q \leq 10$
4. (6 points) Les niveaux de compétence ne dépassent jamais 1.
5. (9 points) Les niveaux de compétence ne dépassent jamais 500.
6. (12 points) $x[i] = 1$ pour chaque $1 \leq i \leq Q$.
7. (10 points) Chaque modification modifie un niveau de compétence d'au plus 1.
8. (18 points) Aucune contrainte supplémentaire.