

Delivery Detour

Mutanguha exploite un service de robots de livraison sur le réseau urbain. Chaque robot démarre en $(0,0)$ et est déjà programmé avec une liste de N instructions $S[0], S[1], \dots, S[N-1]$. Chaque instruction déplace le robot d'une unité : de (x,y) , l'instruction $\mathbb{L}$ le déplace à $(x-1,y)$, $\mathbb{R}$ à $(x+1,y)$, $\mathbb{D}$ à $(x,y-1)$ et $\mathbb{U}$ à $(x,y+1)$.

Juste avant le départ, Mutanguha apprend que le client attend en réalité aux coordonnées (X,Y) . Le micrologiciel du robot n'autorise qu'une seule modification : **l'annulation d'un bloc d'instructions contigu**, éventuellement vide. Le robot exécute ensuite les instructions restantes dans leur ordre initial. Le fabricant facture des frais d'annulation pour chaque instruction annulée, en fonction de sa direction : C_L pour $\mathbb{L}$, C_R pour $\mathbb{R}$, C_D pour $\mathbb{D}$ et C_U pour $\mathbb{U}$.

Aidez Mutanguha à trouver les frais totaux minimums pour que le robot s'arrête à (X,Y) , ou déterminez qu'aucun bloc d'annulations n'atteint ce résultat.

Implementation Details

Vous devez mettre en œuvre la procédure suivante :

```
int64 minimum_fee(string S, int32 X, int32 Y, int32 CL, int32 CR,
                  int32 CD, int32 CU)
```

- S : un string de longueur N sur les caractères $\mathbb{LRDU}$, les instructions programmées.
- X, Y : l'emplacement du client
- CL, CR, CD, CU : les frais d'annulation d'une seule instruction $\mathbb{L}, \mathbb{R}, \mathbb{D}, \mathbb{U}$, respectivement.
- Cette procédure est appelée une seule fois.
- Cette procédure doit renvoyer les frais totaux minimums afin que le robot s'arrête à (X,Y) , ou à -1 si cela est impossible.

Constraints

- $1 \leq N \leq 200\,000$
- $-N \leq X, Y \leq N$
- $0 \leq C_L, C_R, C_D, C_U \leq 10^9$

Subtasks

Subtask	Score	Additional Constraints
1	15	$N \leq 2\,000$
2	10	$C_L = C_R = C_D = C_U = 0$
3	20	S ne contient que R et U
4	20	S ne contient que L et R , et $Y = 0$
5	15	$C_L = C_R = C_D = C_U = 1$
6	20	No additional constraints.

Exemples

Exemple 1

Considérons l'appel suivant :

```
minimum_fee("RUURLLL", -1, 0, 2, 3, 3, 4)
```

L'exécution des sept instructions déplace le robot vers $(-1, 2)$, ce qui manque le client situé à $(-1, 0)$.

Si Mutanguha annule le bloc UU , le robot exécute R, R, L, L, L et atteint la destination cible aux coordonnées $(-1, 0)$. Les frais pour cette annulation sont de $C_U + C_U = 8$.

Bien que l'annulation du bloc le plus long $UURL$ place également le robot en $(-1, 0)$, ses frais totaux sont plus élevés : $4 + 4 + 3 + 2 = 13$ \$. Comme aucune séquence d'annulation ne permet d'atteindre la destination correcte à un coût inférieur à 8 \$, `minimum_fee` devrait renvoyer 8 \$.

Exemple 2

Considérons l'appel suivant :

```
minimum_fee("RR", 0, 1, 1, 1, 1, 1)
```

Aucun bloc d'annulations ne peut arrêter le robot à la position $(0, 1)$, car il ne se déplace jamais verticalement. Par conséquent, `minimum_fee` devrait renvoyer -1 .

Exemple 3

Considérons l'appel suivant :

```
minimum_fee("LU", -1, 1, 7, 7, 7, 7)
```

Le robot s'arrête déjà à $(-1, 1)$, donc Mutanguha devrait annuler le bloc vide et ne rien payer. Par conséquent, `minimum_fee` devrait renvoyer 0 \$.

Sample Grader

Input format:

```
N X Y
CL CR CD CU
S
```

Output format:

```
F
```

Ici, F désigne la valeur de retour de `minimum_fee`.