

נילוס

אתם רוצים להעביר N חבילות דרך הנילוס. החבילות ממוספרות מ-0 עד $N - 1$. המשקל של החבילה ה- i הוא $W[i]$ ($0 \leq i < N$).

על מנת להעביר את החבילות, אתם משתמשים בסירות ייעודיות. כל סירה יכולה לשאת לכל היותר שתי חבילות.

- אם אתם מחליטים לשים חבילה אחת בסירה, משקל החבילה יכול להיות שרירותי.
- אם אתם רוצים לשים שתי חבילות באותה סירה, אתם חייבים לוודא שהסירה מאוזנת. ספציפית, אתם יכולים לשלוח את החבילות p ו- q ($0 \leq p < q < N$) באותה הסירה רק אם ההפרש בין המשקלים שלהן הוא לכל היותר D , כלומר $|W[p] - W[q]| \leq D$.

בשביל להעביר חבילה, עליכם לשלם מחיר התלוי במספר החבילות הנמצאות באותה סירה. המחיר להעברת החבילה ה- i ($0 \leq i < N$) הוא:

- $A[i]$, אם אתם שמים את החבילה לבד בסירה, או
- $B[i]$, אם אתם שמים אותה בסירה יחד עם איזושהי חבילה אחרת.

שימו לב שבמקרה השני, עליכם לשלם עבור שתי החבילות שבסירה. כלומר, אם אתם מחליטים לשלוח את החבילות p ו- q ($0 \leq p < q < N$) באותה סירה, עליכם לשלם $B[p] + B[q]$.

לשלוח חבילה בסירה לבד תמיד יהיה יותר יקר מאשר לשלוח אותה יחד עם חבילה אחרת באותה סירה, ולכן $B[i] < A[i]$ לכל i שמקיים $0 \leq i < N$.

לרוע המזל, הנהר מאוד בלתי צפוי והערך של D משתנה באופן תדיר. עליכם לענות על Q שאלות הממוספרות מ-0 עד $Q - 1$. השאלות מתוארות ע"י מערך E באורך Q . התשובה לשאלתה ה- j ($0 \leq j < Q$) היא המחיר הכולל המינימלי שיש לשלם על מנת להעביר את כל N החבילות, כאשר ערכו של D הוא $E[j]$.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
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 : מערכים של מספרים שלמים באורך N , המתארים את משקלי החבילות ואת מחירי ההובלה שלהן.
- E : מערך של מספרים שלמים באורך Q המתאר את ערכו של D בכל שאלתה.

- על הפונקציה להחזיר מערך R של Q מספרים שלמים, המכיל את המחיר הכולל המינימלי של שליחת החבילות, כאשר $R[j]$ הוא המחיר כאשר ערכו של D הוא $E[j]$ (לכל j שמקיים $0 \leq j < Q$).
- פונקציה זו תיקרא פעם אחת בדיוק בכל טסט.

מגבלות

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq i < N$ עבור כל i שמקיים $1 \leq W[i] \leq 10^9$
- $0 \leq i < N$ עבור כל i שמקיים $1 \leq B[i] < A[i] \leq 10^9$
- $0 \leq j < Q$ עבור כל j שמקיים $1 \leq E[j] \leq 10^9$

תתי משימות

מגבלות נוספות	ניקוד	תת משימה
$Q \leq 5; N \leq 2000; 0 \leq i < N$ עבור כל i שמקיים $W[i] = 1$	6	1
$Q \leq 5; 0 \leq i < N$ עבור כל i שמקיים $W[i] = i + 1$	13	2
$Q \leq 5; 0 \leq i < N$ עבור כל i שמקיים $B[i] = 1$ וגם $A[i] = 2$	17	3
$Q \leq 5; N \leq 2000$	11	4
$Q \leq 5$	20	5
$0 \leq i < N$ עבור כל i שמקיים $B[i] = 1$ וגם $A[i] = 2$	15	6
ללא מגבלות נוספות.	18	7

דוגמה

הביטו בקריאה הבאה.

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

בדוגמה זו יש לנו $N = 5$ חבילות ו- $Q = 3$ שאילתות.

בשאילתה הראשונה, $D = 5$. אתם יכולים לשלוח את החבילות 0 ו-3 באותה סירה (מכיוון ש- $|15 - 10| \leq 5$) ואת שאר החבילות בסירות נפרדות. כך מתקבל המחיר הנמוך ביותר לשליחת כל החבילות, שהוא $1 + 4 + 5 + 3 + 3 = 16$.

בשאילתה השנייה, $D = 9$. אתם יכולים לשלוח את החבילות 0 ו-1 בסירה אחת (מכיוון ש- $|15 - 12| \leq 9$) ואת החבילות 2 ו-3 בסירה אחת (מכיוון ש- $|2 - 10| \leq 9$). את החבילה שנשארה אפשר לשלוח בסירה משלה. כך

מתקבל המחיר הנמוך ביותר לשליחת כל החבילות, שהוא $1 + 2 + 2 + 3 + 3 = 11$.

בשאלתה האחרונה, $D = 1$. אתם צריכים לשלוח כל חבילה בסירה משלה. כך מתקבל המחיר הנמוך ביותר לשליחת כל החבילות, שהוא $5 + 4 + 5 + 6 + 3 = 23$.

לכן, על הפונקציה להחזיר $[16, 11, 23]$.

גריידר לדוגמה

פורמט קלט:

```
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]
```

פורמט פלט:

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

כאשר S הוא אורך המערך R שהוחזר ע"י הפונקציה `calculate_costs`.