

## קפיצות ביערות הגשם

ביערות הגשם הטרופיים של סומטרה, ישנם  $N$  עצים בשורה הממוספרים מ-0 עד  $N - 1$  משמאל לימין. לכל העצים גבהים יחודיים, כשהעץ  $i$  הוא בגובה  $H[i]$ .

פאק דנגלק מאמן אורנגאוטנית לקפוץ מעץ לעץ. בקפיצה בודדת, האורנגאוטנית יכולה לקפוץ מצמרת עץ מסוים לצמרת העץ הקרוב ביותר, משמאל או מימין, שגובהו גבוה מהעץ שהיא כרגע נמצאת עליו. פורמלית, אם האורנגאוטנית כרגע נמצאת בעץ  $x$ , אז היא יכולה לקפוץ לעץ  $y$  אם ורק אם אחד מהתנאים הבאים מתקיים:

- $y$  הוא השלם האי-שלילי הגדול ביותר שקטן מ- $x$  כך ש- $H[y] > H[x]$ ; או
- $y$  הוא השלם האי-שלילי הקטן ביותר שגדול מ- $x$  כך ש- $H[y] > H[x]$ .

לפאק דנגלק יש  $Q$  תוכניות קפיצה, שכל אחת ניתן לייצג כארבעה מספרים שלמים  $A, B, C$ , ו- $D$  ( $A \leq B < C \leq D$ ). עבור כל אחת מהתוכניות, פאק דנגלק מעוניין לדעת האם ייתכן שהאורנגאוטנית תתחיל מעץ  $s$  ( $A \leq s \leq B$ ) כלשהו ותסיים בעץ  $e$  ( $C \leq e \leq D$ ) כלשהו על ידי רצף של קפיצות. אם זה אפשרי, פאק דנגלק רוצה לדעת את מספר הקפיצות הקטן ביותר שהאורנגאוטנית צריכה לעשות לתוכנית זו.

### פרטי מימוש

עליכם לממש את הפונקציות הבאות:

```
void init(int N, int[] H)
```

- $N$ : מספר העצים.
- $H$ : מערך באורך  $N$ , כש- $H[i]$  הוא גובה העץ  $i$ .
- פונקציה זו נקראת בדיוק פעם אחת, לפני כל הקריאות ל-`minimum_jumps`.

```
int minimum_jumps(int A, int B, int C, int D)
```

- $A, B$ : טווח העצים שהאורנגאוטנית חייבת להתחיל בו.
- $C, D$ : טווח העצים שהאורנגאוטנית חייבת לסיים בו.
- על פונקציה זו להחזיר את מספר הקפיצות הקטן ביותר הדרוש על מנת לבצע את התוכנית, או  $-1$  אם בלתי אפשרי לעשות זאת.
- פונקציה זו נקראת בדיוק  $Q$  פעמים.

### דוגמה

הביטו בקריאה הבאה:

```
init(7, [3, 2, 1, 6, 4, 5, 7])
```

לאחר שהאתחול נעשה, הביטו בקריאה הבאה:

```
minimum_jumps(4, 4, 6, 6)
```

משמעותה שהאורנגאוטנית חייבת להתחיל בעץ 4 (שגובהו 4) ולסיים בעץ 6 (שגובהו 7). דרך אחת להשיג את מספר הקפיצות הקטן ביותר היא תחילה לקפוץ לעץ 3 (שגובהו 6), ואחר כך לקפות לעץ 6. דרך אחרת היא לקפוץ לעץ 5 (שגובהו 5), ואז לקפוץ לעץ 6. לכן, על הפונקציה `minimum_jumps` להחזיר 2.

הביטו בקריאה אפשרית נוספת:

```
minimum_jumps(1, 3, 5, 6)
```

משמעותה שהאורנגאוטנית חייבת להתחיל בעץ 1 (שגובהו 2), בעץ 2 (שגובהו 1) או בעץ 3 (שגובהו 6), ולסיים בעץ 5 (שגובהו 5) או בעץ 6 (שגובהו 7). הדרך היחידה להשיג את מספר הקפיצות הקטן ביותר היא להתחיל בעץ 3, ואז לקפוץ לעץ 6 באמצעות קפיצה אחת בלבד. לכן, על הפונקציה `minimum_jumps` להחזיר 1.

הביטו בקריאה אפשרית נוספת:

```
minimum_jumps(0, 1, 2, 2)
```

משמעותה שהאורנגאוטנית חייבת להתחיל בעץ 0 (שגובהו 3), או בעץ 1 (שגובהו 2), ולסיים בעץ 2 (שגובהו 1). מאחר שעץ 2 הוא העץ הנמוך ביותר, בלתי אפשרי להגיע אליו מאף עץ שגובהו ממנו. לכן, על הפונקציה `minimum_jumps` להחזיר -1.

## מגבלות

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq H[i] \leq N$  (לכל  $0 \leq i \leq N - 1$ )
- $H[i] \neq H[j]$  (לכל  $0 \leq i < j \leq N - 1$ )
- $0 \leq A \leq B < C \leq D \leq N - 1$

## תת משימות

1. (4 נקודות)  $H[i] = i + 1$  (לכל  $0 \leq i \leq N - 1$ )
2. (8 נקודות)  $N \leq 200, Q \leq 200$
3. (13 נקודות)  $N \leq 2000, Q \leq 2000$
4. (12 נקודות)  $Q \leq 5$
5. (23 נקודות)  $A = B, C = D$
6. (21 נקודות)  $C = D$

## גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1:  $N$   $Q$
- שורה 2:  $H[0]$   $H[1]$  ...  $H[N - 1]$
- שורה  $3 + i$ :  $A$   $B$   $C$   $D$ :  $(0 \leq i \leq Q - 1)$  עבור הקריאה ה- $i$  ל-`minimum_jumps`

הגריידר לדוגמה מדפיס התשובות שלכם בפורמט הבא:

- שורה  $1 + i$ :  $(0 \leq i \leq Q - 1)$  ערך החזרה של הקריאה ה- $i$  ל-`minimum_jumps`