

## Roller Coaster Railroad

אנה (Anna) עובדת בפארק שעשועים והיא אחראית על בניית מסילה (railroad) עבור רכבת הרים חדשה. היא כבר תכננה  $n$  איזורים (special sections) (הממוספרים מ-0 עד  $n - 1$ ) שמשפיעים על מהירות הרכבת. כעת היא צריכה לחבר את האיזורים יחד ולהציע תכנון סופי לרכבת ההרים. בבעיה זו הניחו שאורך הרכבת הוא אפס.

לכל  $i$  בין 0 ל- $n - 1$ , כולל, לאיזור  $i$  יש שתי תכונות:

- כאשר נכנסים לאיזור, יש הגבלת מהירות: מהירות הרכבת חייבת להיות קטנה או שווה ל- $s_i$  קמ"ש (קילומטרים לשעה).
- כאשר עוזבים את האיזור, מהירות הרכבת היא בדיוק  $t_i$  קמ"ש, ללא קשר למהירות שבה הרכבת נכנסה לאיזור.

מסילת הרכבת הסופית היא מסלול קווי שכולל את  $n$  האיזורים בסדר כלשהו. כל אחד מהאיזורים מופיע פעם אחת בדיוק. איזורים עוקבים מחוברים במקטע מסילתי (track). אנה צריכה לבחור את הסדר של  $n$  האיזורים ולהחליט על אורכי המקטעים המסילתיים. אורך של מקטע מסילתי נמדד במטרים ועשוי להיות כל מספר שלם אי-שלילי (מותר אפס).

כל מטר של מקטע מסילתי בין שני איזורים מאט את הרכבת ב-1 קמ"ש. בתחילת הנסיעה, הרכבת נכנסת לאיזור הראשון בסדר שבחרה אנה, במהירות של 1 קמ"ש.

התכנון הסופי צריך לקיים את הדרישות הבאות:

- הרכבת לא חורגת ממגבלות המהירות כשהיא נכנסת לאיזורים השונים.
- מהירות הרכבת חיובית בכל רגע.

בכל תת המשימות, למעט תת משימה מספר 3, עליכם למצוא את האורך הכולל המינימלי האפשרי של המקטעים המסילתיים. בתת משימה מספר 3 עליכם רק לבדוק האם קיים תכנון שעונה על הדרישות ושבנו כל מקטע מסילתי הוא באורך אפס.

### פרטי מימוש

עליכם לממש את הפונקציה הבאה (שיטה):

• `int64 plan_roller_coaster(int[] s, int[] t)`

- $s$ : מערך בגודל  $n$ , מהירויות הכניסה המקסימליות.
- $t$ : מערך בגודל  $n$ , מהירויות היציאה.
- בכל תת המשימות למעט תת משימה מספר 3, הפונקציה צריכה להחזיר את האורך הכולל המינימלי האפשרי של המקטעים המסילתיים. בתת משימה מספר 3, הפונקציה צריכה להחזיר 0 אם קיים תכנון שעונה על הדרישות ושבנו כל מקטע מסילתי הוא באורך אפס, או מספר שלם חיובי כלשהו אם לא קיים תכנון כזה.

עבור שפת C חתימת הפונקציה שונה מעט :

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
  - $n$  : מספר האיברים ב-  $s$  ו-  $t$  (כלומר, מספר האיזורים).
  - שאר הפרמטרים זהים למתואר לעיל.

## דוגמה

`plan_roller_coaster ([1, 4, 5, 6], [7, 3, 8, 6])`

בדוגמה זו יש ארבעה איזורים. הפתרון הטוב ביותר הוא לבנות אותם בסדר הבא משמאל לימין:  $0, 3, 1, 2$  ולחבר ביניהם עם מקטעים מסילתיים באורכים  $1, 2, 0$  בהתאמה. הרכבת נוסעת באופן הבא :

- בהתחלה מהירות הרכבת היא  $1$  קמ"ש.
- הרכבת מתחילה את נסיעתה בכניסה לאיזור  $0$ .
- הרכבת יוצאת מאיזור  $0$  במהירות של  $7$  קמ"ש.
- כעת יש מקטע מסילתי באורך  $1$  מטר. כשהרכבת מגיעה לקצה המקטע הזה, מהירותה היא  $6$  קמ"ש.
- הרכבת נכנסת לאיזור  $3$  במהירות של  $6$  קמ"ש ויוצאת ממנו באותה מהירות.
- לאחר עזיבתה את איזור  $3$ , הרכבת נוסעת על מקטע מסילתי באורך  $2$  מטר. מהירותה יורדת ל- $4$  קמ"ש.
- הרכבת נכנסת לאיזור  $1$  במהירות  $4$  קמ"ש ועוזבת אותו במהירות  $3$  קמ"ש.
- מיד לאחר איזור  $1$ , הרכבת נכנסת לאיזור  $2$ .
- הרכבת עוזבת את איזור  $2$ . מהירותה הסופית היא  $8$  קמ"ש.

הפונקציה צריכה להחזיר את האורך הכולל של המקטעים המסילתיים בין האיזורים:  $1 + 2 + 0 = 3$

## תת משימות

בכל תת המשימות מתקיים  $1 \leq s_i \leq 10^9$  וגם  $1 \leq t_i \leq 10^9$ .

1. (11 נקודות):  $2 \leq n \leq 8$ .
2. (23 נקודות):  $2 \leq n \leq 16$ .
3. (30 נקודות):  $2 \leq n \leq 200\,000$ . בתת משימה זו התוכנית צריכה רק לבדוק האם התשובה היא אפס או לא. אם התשובה אינה אפס, כל מספר שלם חיובי נחשב כתשובה נכונה.
4. (36 נקודות):  $2 \leq n \leq 200\,000$ .

## גריידר לדוגמה

הגריידר קורא קלט בפורמט הבא :

- שורה ראשונה : המספר השלם  $n$ .
- לכל  $i$  בין  $0$  ל- $n - 1$ , שורה  $i + 2$  מכילה את המספר השלם  $s_i$  ואחריו  $t_i$ .