

## משחק צינוקים

רוברט מתכנן משחק מחשב חדש. המשחק מכיל גיבור אחד,  $n$  יריבים ו- $n + 1$  צינוקים. היריבים ממוספרים מ-0 עד  $n - 1$  והצינוקים ממוספרים מ-0 עד  $n$ . היריב ה- $i$  ( $0 \leq i \leq n - 1$ ) ממוקם בצינוק ה- $i$  ובעל עוצמה  $s[i]$ . אין יריב בצינוק ה- $n$ .

הגיבור מתחיל בכניסה לצינוק ה- $x$ , עם עוצמה  $z$ . בכל פעם שהגיבור נכנס לצינוק כלשהו  $i$  ( $0 \leq i \leq n - 1$ ), הוא מתמודד עם היריב ה- $i$  ואחת מהתוצאות הבאות מתרחשת:

- אם העוצמה של הגיבור גדולה או שווה לעוצמה של היריב,  $s[i]$ , הגיבור מנצח. זה גורם לעוצמה שלו **לגדול** ב- $s[i]$  ( $s[i] \geq 1$ ). במקרה זה הגיבור נכנס לצינוק  $w[i]$  לאחר מכן ( $w[i] > i$ ).

- אחרת, הגיבור מפסיד. זה גורם לעוצמה שלו **לגדול** ב- $p[i]$  ( $p[i] \geq 1$ ). במקרה הזה הגיבור נכנס לצינוק  $l[i]$  לאחר מכן.

שימו לב כי  $p[i]$  עשוי להיות קטן מ- $s[i]$ , שווה לו, או גדול ממנו. כמו כן,  $l[i]$  עשוי להיות קטן מ- $i$ , שווה לו, או גדול ממנו. ללא קשר לתוצאות העימות, היריב נשאר בצינוק ה- $i$  ושומר על העוצמה  $s[i]$ .

המשחק מסתיים כאשר הגיבור נכנס לצינוק ה- $n$ . ניתן להראות כי המשחק מסתיים אחרי מספר סופי של עימותים, ללא קשר לצינוק ההתחלתי של הגיבור או לעוצמתו.

רוברט ביקש מכם לבדוק את המשחק שלו ע"י הרצת  $q$  סימולציות. לכל סימולציה, רוברט מגדיר צינוק התחלה  $x$  ועוצמה התחלתית  $z$ . משימתכם היא למצוא, לכל סימולציה, את העוצמה של הגיבור כאשר המשחק מסתיים.

## פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
void init(int n, int[] s, int[] p, int[] w, int[] l)
```

- $n$ : מספר היריבים.
- $s, p, w, l$ : מערכים בגודל  $n$ . לכל  $0 \leq i \leq n - 1$ :
  - $s[i]$ : העוצמה של היריב ה- $i$ . זוהי גם העוצמה שהגיבור מקבל אחרי ניצחון מול היריב ה- $i$ .
  - $p[i]$ : העוצמה שהגיבור מקבל אחרי הפסד מול היריב ה- $i$ .
  - $w[i]$ : הצינוק שהגיבור נכנס אליו אחרי ניצחון מול היריב ה- $i$ .
  - $l[i]$ : הצינוק שהגיבור נכנס אליו אחרי הפסד מול היריב ה- $i$ .
- הפונקציה הזו נקראת בדיוק פעם אחת, לפני כל הקריאות ל-simulate (ראו מטה).

```
int64 simulate(int x, int z)
```

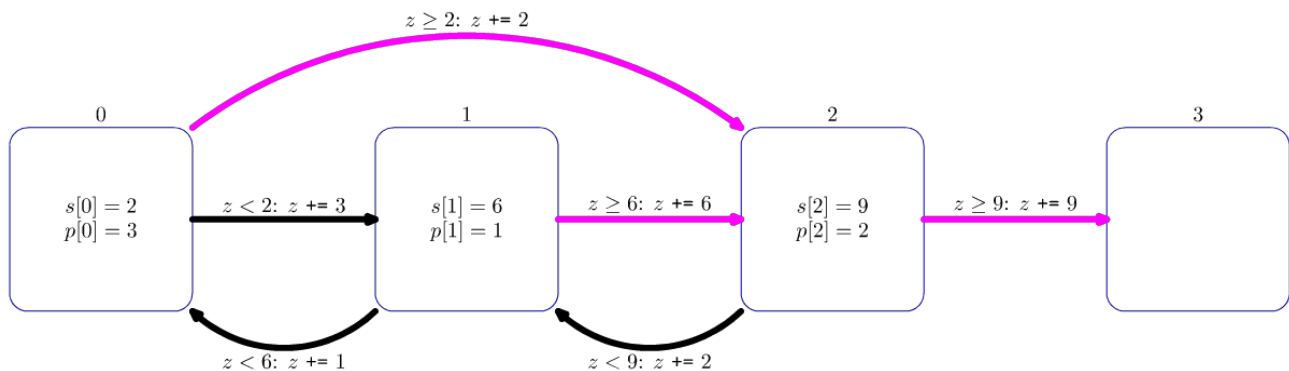
- $x$ : הצינוק שאליו הגיבור נכנס בהתחלה.

- $z$  : העוצמה ההתחלתית של הגיבור.
- הפונקציה הזו צריכה להחזיר את העוצמה של הגיבור כשהמשחק מסתיים, בהנחה שהגיבור מתחיל את המשחק ע"י כניסה לצינוק  $x$  עם עוצמה  $z$ .
- הפונקציה הזו תיקרא בדיוק  $q$  פעמים.

## דוגמה

הביטו בקריאה הבאה:

```
init(3, [2, 6, 9], [3, 1, 2], [2, 2, 3], [1, 0, 1])
```



הסרטוט לעיל מתאר את הקריאה. כל ריבוע מראה צינוק. עבור צינוקים 0, 1 ו-2, הערכים  $s[i]$  ו- $p[i]$  מצויינים בתוך הריבועים. חצים ורודים מסמנים לאן הגיבור זז לאחר ניצחון של עימות, וחצים שחורים מסמנים לאן הגיבור זז לאחר הפסד.

נניח כי הגריידר קורא ל-`simulate(0, 1)`.

המשחק מתבצע כך:

צינוק	עוצמת הגיבור לפני העימות	תוצאה
0	1	הפסד
1	4	הפסד
0	5	ניצחון
2	7	הפסד
1	9	ניצחון
2	15	ניצחון
3	24	המשחק מסתיים

לפיכך, הפונקציה צריכה להחזיר 24.

נניח כי הגריידר קורא ל-`simulate(2, 3)`.

המשחק מתבצע כך:

צינוק	עוצמת הגיבור לפני העימות	תוצאה
2	3	הפסד
1	5	הפסד
0	6	ניצחון
2	8	הפסד
1	10	ניצחון
2	16	ניצחון
3	25	המשחק מסתיים

לפיכך, הפונקציה צריכה להחזיר 25.

## מגבלות

- $1 \leq n \leq 400\,000$
- $1 \leq q \leq 50\,000$
- $(0 \leq i \leq n - 1)$  לכל  $1 \leq s[i], p[i] \leq 10^7$
- $(0 \leq i \leq n - 1)$  לכל  $0 \leq l[i], w[i] \leq n$
- $(0 \leq i \leq n - 1)$  לכל  $w[i] > i$
- $0 \leq x \leq n - 1$
- $1 \leq z \leq 10^7$

## תת משימות

1. (11 נקודות)  $s[i], p[i] \leq 10\,000$ ,  $q \leq 100$ ,  $n \leq 50\,000$  (לכל  $0 \leq i \leq n - 1$ )
2. (26 נקודות)  $s[i] = p[i]$  (לכל  $0 \leq i \leq n - 1$ )
3. (13 נקודות)  $n \leq 50\,000$ , כל היריבים בעלי עוצמה זהה. במילים אחרות,  $s[i] = s[j]$  לכל  $0 \leq i, j \leq n - 1$ .
4. (12 נקודות)  $n \leq 50\,000$ , יש לכל היותר 5 ערכים שונים מבין ערכי  $s[i]$ .
5. (27 נקודות)  $n \leq 50\,000$
6. (11 נקודות) ללא מגבלות נוספות.

## גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1:  $n$
- שורה 2:  $s[0] \ s[1] \ \dots \ s[n - 1]$
- שורה 3:  $p[0] \ p[1] \ \dots \ p[n - 1]$
- שורה 4:  $w[0] \ w[1] \ \dots \ w[n - 1]$
- שורה 5:  $l[0] \ l[1] \ \dots \ l[n - 1]$
- שורה  $6 + i$ :  $(0 \leq i \leq q - 1)$   $x \ z$  עבור הקריאה ה- $i$  ל-simulate.

הגרייטר לדוגמה מדפיס את תשובותיכם בפורמט הבא:

• שורה  $1 + i$  ( $0 \leq i \leq q - 1$ ): ערך ההחזרה של הקריאה ה- $i$  ל-`simulate`.