APIO 2016
7th May 2016
Republic of Korea

gap
Language: he_IL

Gap

יש N מספרים שלמים אי-שליליים $a_1, a_2, \ldots, a_N$ שמקיימים את האי-שוויון $0 \leq a_1 < a_2 < \cdots < a_N \leq 10^{18}$. ג'יהק (Jeehak) רוצה לדעת מהו הערך הגדול ביותר האפשרי של $a_{i+1} - a_i$, כאשר i בין 1 ל-$N-1$. המספרים אינם נתונים ישירות לתוכנית של ג'יהק, אלא נגישים דרך פונקציה מיוחדת. פרטים נוספים תמצאו בחלק המימוש המתאים לשפת התכנות שלכם.

משימה

עזרו לג'יהק לממש פונקציה שתחזיר את הערך הגדול ביותר האפשרי של $a_{i+1} - a_i$ כאשר i בין 1 ל-$N-1$.

מימוש ב-C ו-C++

עליכם לממש פונקציה אחת findGap(T, N) אשר מקבלת את הפרמטרים הבאים ומחזירה משתנה מטיפוס long long:

- המשתנה T - המספר של תת המשימה הנוכחית (1 או 2)
- המשתנה N - כמות המספרים הנתונים

לפונקציה findGap שלכם מותר לקרוא לפונקציה MinMax(s, t, &mn, &mx) כאשר שני הפרמטרים הראשונים s ו-t הם משתנים מספריים מסוג long long, ושני הפרמטרים האחרונים mn& ו-mx& הם מצביעים למשתנים מספריים מסוג long long, כלומר mn ו-mx משתנים מספריים מסוג long long. כאשר MinMax(s, t, &mn, &mx) מסיימת, המשתנה mn יהיה שווה לערך ה-a_i הקטן ביותר אשר גדול או שווה ל-s, והמשתנה mx יהיה שווה לערך ה-a_j הגדול ביותר אשר קטן או שווה ל-t. אם אין מספרים נתונים בין s ל-t (כולל), אז גם mn וגם mx יהיו שווים ל--1. הערך של s חייב להיות קטן או שווה ל-t בכל קריאה ל-MinMax. אם תנאי זה אינו מתקיים, התוכנית תסתיים עם ערך החזרה שונה מ-0.

מימוש בפסקל

עליכם לממש פונקציה אחת findGap(T, N) אשר מקבלת את הפרמטרים הבאים ומחזירה משתנה מטיפוס Int64:

- המשתנה T - המספר של תת המשימה הנוכחית (1 או 2) (מסוג Integer)
- המשתנה N - כמות המספרים הנתונים (מסוג LongInt)

לפונקציה findGap שלכם מותר לקרוא לפונקציה MinMax(s, t, mn, mx) כאשר שני הפרמטרים הראשונים s ו-t הם משתנים מספריים מסוג Int64, ושני הפרמטרים האחרונים mn ו-mx הם **רפרנסים** למשתנים מספריים מסוג Int64, כלומר mn ו-mx משתנים מספריים מסוג Int64. כאשר MinMax(s, t, mn, mx) מסיימת, המשתנה mn יהיה שווה לערך ה-a_i הקטן ביותר אשר גדול או שווה ל-s, והמשתנה mx יהיה שווה לערך ה-a_j הגדול ביותר אשר קטן או שווה ל-t. אם אין מספרים נתונים בין s ל-t (כולל), אז גם mn וגם mx יהיו שווים ל--1. הערך של s חייב להיות קטן או שווה ל-t בכל קריאה ל-MinMax. אם תנאי זה אינו מתקיים, התוכנית תסתיים.

פרטי מימוש כלליים

בנוסף לדרישות הרגילות (התוכנית צריכה לעמוד בהגבלות זמן וזיכרון, לא לקרוס, וכו') ההגשה שלכם חייבת לעמוד בתנאים הבאים:

- הפונקציה findGap חייבת להחזיר את התשובה הנכונה.
- על כל קריאה ל-MinMax משלמים מחיר מסוים, והמחיר הכולל M חייב לא לחרוג מהמותר (ראו פרטים בחלק על ניקוד).

דוגמה עבור C, ++C

נביט במקרה של $N = 4$ כאשר $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$.
הפונקציה findGap יכולה להסיק את התשובה הנכונה, 3, על ידי הקריאות הבאות ל-MinMax:

- הקריאה MinMax(1, 2, &mn, &mx) תציב ב-mn וגם ב-mx את הערך 2.
- הקריאה MinMax(3, 7, &mn, &mx) תציב ב-mn את הערך 3, וב-mx את הערך 6.
- הקריאה MinMax(8, 9, &mn, &mx) תציב ב-mn וגם ב-mx את הערך 8.

דוגמה עבור פסקל

נביט במקרה של $N = 4$ כאשר $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$.
הפונקציה findGap יכולה להסיק את התשובה הנכונה, 3, על ידי הקריאות הבאות ל-MinMax:

- הקריאה MinMax(1, 2, mn, mx) תציב ב-mn וגם ב-mx את הערך 2.
- הקריאה MinMax(3, 7, mn, mx) תציב ב-mn את הערך 3, וב-mx את הערך 6.
- הקריאה MinMax(8, 9, mn, mx) תציב ב-mn וגם ב-mx את הערך 8.

ניקוד

בכל תת המשימות מתקיים $2 \leq N \leq 100,000$.

תת משימה 1 (30 נקודות): כל קריאה ל-MinMax תוסיף 1 למחיר הכולל M. תקבלו ניקוד מלא עבור תת המשימה אם יתקיים $M \leq \frac{N+1}{2}$ בכל אחד מהקלטים.

תת משימה 2 (70 נקודות): נגדיר את k להיות כמות המספרים הנתונים אשר גדולים או שווים ל-s וקטנים או שווים ל-t בקריאה כלשהי ל-MinMax. קריאה כזאת ל-MinMax תוסיף $k + 1$ למחיר M. הניקוד הסופי יחושב כך: הניקוד של תת המשימה הוא הניקוד המינימלי שתקבלו מבין הקלטים שבתוכה. עבור כל קלט, הניקוד יהיה 70 אם $M \leq 3N$, אחרת הניקוד יהיה $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$.

ניסוי

קיים grader לדוגמה, שאפשר להוריד מהמערכת, והוא קורא את כל הנתונים מ-standard input. השורה הראשונה בקלט שלו צריכה להכיל שני מספרים, מספר תת המשימה T ואחריו N. השורה הבאה צריכה להכיל את N המספרים בסדר עולה. ה-grader יכתוב ל-standard output את הערך שהחזירה הפונקציה findGap ואת הערך M בהתאם לתת המשימה.
הקלט הבא מתאים לדוגמה למעלה:

```
2 4
2 3 6 8
```