



International Olympiad in Informatics 2015

26th July - 2nd August 2015

Almaty, Kazakhstan

Day 2

horses

Language: he-IL

Horses

מנצור (Mansur) עוסק בגידול סוסים, כמו אבותיו הקדומים. לפני N שנים, מנצור היה סתם איש צעיר (בקזחית: dzhigit) והיה לו בדיוק סוס אחד. מאז הוא גידל הרבה סוסים והתעשר וכיום הוא איש עשיר (בקזחית: bai).

נמספר את השנים מאז שמנצור התחיל את עסקיו מ-0 עד $N-1$, בסדר כרונולוגי (כלומר, שנה $N-1$ היא השנה האחרונה). בכל שנה, מספר הסוסים בעדר של מנצור גדל באופן שתלוי במזג האוויר של אותה שנה. כדי לשקף זאת, לשנה i מתאים מקדם שלם חיובי $X[i]$. אם בתחילת שנה i יש בעדר h סוסים, בסוף השנה יש בעדר $h \cdot X[i]$ סוסים.

בסוף כל שנה, ורק בסוף השנה, מנצור יכול למכור סוסים. לשנה i מתאים מספר שלם חיובי $Y[i]$ שהוא המחיר בו ניתן למכור סוס בסוף השנה i . למנצור מותר למכור בסוף כל שנה כמה סוסים שהוא רוצה מתוך העדר, כל סוס במחיר $Y[i]$.

מנצור שואל מה סכום הכסף המקסימלי שהוא יכול להרוויח במשך N השנים אילו היה מוכר סוסים בנקודות זמן אופטימליות לצורך הגדלת כמות הכסף שלו ככל האפשר. אתם מתארחים בביתו של מנצור לארוחת ערב, ועליכם לענות על השאלה שלו.

כדי לסבך את העניינים, הזיכרון של מנצור בנוגע לתנאים שהתקיימו בעבר משתנה במהלך הערב. לכן, מנצור מבצע סדרה של M פעולות עדכון. בכל עדכון, מנצור משנה את אחד הערכים $X[i]$ או את אחד הערכים $Y[i]$. לאחר העדכון, הוא מבקש מכם לחשב מחדש את כמות הכסף הגדולה ביותר שהוא יכול היה להרוויח ממכירת הסוסים, הפעם תחת התנאים החדשים. העדכונים שמנצור מבצע מצטברים. כלומר, לאחר כל עדכון, גם העדכונים הקודמים שמנצור ביצע עדיין רלוונטיים. שימו לב שיתכן שאותו $X[i]$ או $Y[i]$ עשוי להתעדכן מספר פעמים.

התשובות לשאלות של מנצור עשויות להיות מספרים גדולים מאוד. כדי שלא תצטרכו לעבוד עם מספרים גדולים, עליכם לענות את התשובות מודולו $10^9 + 7$.

דוגמה

נניח שיש $N = 3$ שנים, עם הנתונים הבאים:

	0	1	2
X	2	1	3
Y	3	4	1

בסיטואציה המתוארת פה, מנצור ירוויח הכי הרבה אם הוא ימכור את שני הסוסים שיהיו לו בסוף שנה 1. התהליך המלא יראה כך:

- בהתחלה יש למנצור סוס אחד.
 - אחרי שנה 0 יש לו $2 \cdot X[0] = 2$ סוסים.
 - אחרי שנה 1 יש לו $2 \cdot X[1] = 2$ סוסים.
 - הוא מוכר את שני הסוסים האלה. הרווח שלו הוא $2 \cdot Y[1] = 8$.
- כעת, נניח שיש $M = 1$ עדכונים. העדכון האחד הוא לשנות את $Y[1]$ ל-2. הנתונים לאחר העדכון הם:

	0	1	2
X	2	1	3
Y	3	2	1

במקרה הזה, אחד הפתרונות האופטימליים הוא למכור סוס אחד בסוף שנה 0 ולמכור עוד שלושה סוסים בסוף שנה 2. התהליך המלא יראה כך:

- בהתחלה, יש למנצור סוס אחד.
- אחרי שנה 0 יש לו $2 = 1 \cdot X[0]$ סוסים.
- הוא מוכר את אחד הסוסים במחיר $Y[0] = 3$. נשאר לו סוס אחד.
- בסוף שנה 1 יש לו $1 = 1 \cdot X[1]$ סוס.
- בסוף שנה 2 יש לו $3 = 1 \cdot X[2]$ סוסים.
- הוא מוכר את שלושת הסוסים ומרוויח $3 \cdot Y[2] = 3$. הרווח הכולל של מנצור במהלך כל השנים הוא $3 + 3 = 6$.

משימה

נתונים לכם N, X, Y ורשימת עדכונים. לפני העדכון הראשון ולאחר כל עדכון, עליכם לחשב את סכום הכסף המקסימלי שמנצור יכול להרוויח על-ידי מכירת הסוסים שלו, מודולו $10^9 + 7$. עליכם לממש את הפונקציות `init`, `updateX` ו-`updateY`.

- הפונקציה `init(N, X, Y)` - ה-`grader` קורא לפונקציה הזאת בהתחלה. הוא קורא לה בדיוק פעם אחת.
 - הפרמטר N : מספר השנים.
 - הפרמטר X : מערך באורך N . עבור $0 \leq i \leq N - 1$, הערך $X[i]$ הוא שיעור הגדילה של עדר הסוסים בשנה i .
 - הפרמטר Y : מערך באורך N . עבור $0 \leq i \leq N - 1$, הערך $Y[i]$ הוא המחיר של סוס בסוף השנה i .
 - שימו לב ש- X ו- Y הם הערכים המקוריים (לפני שמנצור מבצע עדכונים).
 - לאחר שהפונקציה `init` מסתיימת, המערכים X ו- Y נשארים זמינים בזיכרון. מותר לכם לשנות את הערכים שלהם ולקרוא מהם.
 - הפונקציה צריכה להחזיר את סכום הכסף המקסימלי שמנצור יכול להרוויח תחת הערכים הנתונים של X ו- Y , מודולו $10^9 + 7$.
 - הפונקציה `updateX(pos, val)`
 - הפרמטר `pos`: מספר שלם בטווח $0, \dots, N - 1$.
 - הפרמטר `val`: הערך החדש של $X[pos]$.
 - הפונקציה צריכה להחזיר את סכום הכסף המקסימלי שמנצור יכול להרוויח בעקבות העדכון הנוכחי, מודולו $10^9 + 7$.
 - הפונקציה `updateY(pos, val)`
 - הפרמטר `pos`: מספר שלם בטווח $0, \dots, N - 1$.
 - הפרמטר `val`: הערך החדש של $Y[pos]$.
 - הפונקציה צריכה להחזיר את סכום הכסף המקסימלי שמנצור יכול להרוויח בעקבות העדכון הנוכחי, מודולו $10^9 + 7$.
- ניתן להניח שכל הערכים של $X[i]$ ו- $Y[i]$, גם בתחילת הריצה וגם בעקבות העדכונים, הם בין 1 ל- 10^9 (כולל).
- אחרי שה-`grader` קורא ל-`init`, הוא יקרא מספר פעמים ל-`updateX` ול-`updateY`. את מספר הקריאות הכולל ל-`updateX` ול-`updateY` נסמן ב- M .

תת-משימות

subtask	points	N	M	additional constraints
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10,$ $X[0] \cdot X[1] \cdot \dots \cdot X[N-1] \leq 1,000$
2	17	$1 \leq N \leq 1,000$	$0 \leq M \leq 1,000$	none
3	20	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	$X[i] \geq 2$ and $val \geq 2$ for <code>init</code> and <code>updateX</code> correspondingly
4	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 10,000$	none
5	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	none

תכנית grader לדוגמה

ה- grader לדוגמה קורא קלט מהקובץ `horses.in` בפורמט הבא:

- line 1: N
- line 2: $X[0] \dots X[N-1]$
- line 3: $Y[0] \dots Y[N-1]$
- line 4: M
- lines 5,...,M+4: three numbers `type pos val` (`type=1` for `updateX` and `type=2` for `updateY`).

ה- grader לדוגמה מדפיס את הערך שמחזירה `init` ואחריו את הערכים שמחזירות כל הקריאות ל- `updateX` ול- `updateY`.