עקיפה

יש כביש חד-נתיבי וחד-סטרי משדה התעופה בבודפשט למלון פוראש. הכביש באורך L ק"מ.

לאורך IOI 2023, $N+1$ אוטובוסים נוסעים בכביש הזה. האוטובוסים ממוספרים מ-0 עד N. אוטובוס i ($0 \leq i < N$) מתוכנן לצאת משדה התעופה בשנייה ה-$T[i]$ של האירוע, ויכול לנסוע מרחק של 1 ק"מ ב-$W[i]$ שניות. אוטובוס N הוא אוטובוס רזרבי שיכול לנסוע מרחק של 1 ק"מ ב-X שניות. הזמן Y שבו הוא ייצא משדה התעופה עוד לא הוחלט.

אסור לעקוף בכביש באופן כללי, אבל לאוטובוסים מותר לעקוף אחד את השני ב**תחנות מיון**. ישנן M ($M > 1$) תחנות מיון, הממוספרות מ-0 עד $M-1$, במקומים שונים על הכביש. תחנת המיון j ($0 \leq j < M$) ממוקמת במרחק $S[j]$ קילומטרים לאורך הכביש משדה התעופה. תחנות המיון ממויינות בסדר מרחקים עולה משדה התעופה, כלומר, $S[j] < S[j+1]$ לכל $0 \leq j \leq M-2$. תחנת המיון הראשונה היא שדה התעופה והאחרונה היא המלון, כלומר $S[0] = 0$ ו-$S[M-1] = L$.

כל אוטובוס נוסע במהירות מקסימלית אלא אם הוא מגיע לאוטובוס איטי יותר שנוסע מלפניו בכביש, במקרה זה הם מתקבצים ומוכרחים לנסוע במהירות של האוטובוס האיטי יותר, עד שיגיעו לתחנת המיון הבאה. שם, האוטובוסים המהירים יותר יעקפו את האוטובוסים האיטיים יותר.

פורמלית, לכל i ו-j המקיימים $0 \leq i \leq N$ וגם $0 \leq j < M$, הזמן $t_{i,j}$ (בשניות) שבו אוטובוס i **מגיע** לתחנת המיון j מוגדר כדלקמן. נגדיר $t_{i,0} = T[i]$ לכל $0 \leq i < N$, ונגדיר $t_{N,0} = Y$. לכל j המקיים $0 < j < M$:

- נגדיר את **זמן ההגעה הצפוי** (בשניות) של אוטובוס i לתחנת המיון j, המסומן ב- $e_{i,j}$, להיות הזמן שבו אוטובוס i היה מגיע לתחנת המיון j, אם הוא היה נוסע במהירות המקסימלית שלו מאז הזמן שהגיע לתחנת המיון $j-1$. כלומר, נגדיר
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ לכל $0 \leq i < N$, ו-
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- אוטובוס i מגיע לתחנת המיון j בזמן *המקסימלי* מבין זמני ההגעה הצפויים של אוטובוס i ושל כל אוטובוס אחר שהגיע לתחנת המיון $j-1$ לפני אוטובוס i. פורמלית, נגדיר $t_{i,j}$ להיות המקסימום של $e_{i,j}$ ושל כל $e_{k,j}$ שעבורו $0 \leq k \leq N$ וגם $t_{k,j-1} < t_{i,j-1}$.

מארגני ה-IOI רוצים לתכנן את זמן היציאה של האוטובוס הרזרבי (אוטובוס N). משימתכם היא לענות על Q שאילתות של המארגנים, שהן מהצורה הבאה: בהינתן הזמן Y (בשניות) שבו האוטובוס הרזרבי אמור לצאת משדה התעופה, באיזו שעה הוא יגיע למלון?

פרטי מימוש

משימתכם היא לממש את הפונקציות הבאות.

```
void init(int L, int N, int64[] T, int[] W, int X, int M, int[] S)
```

- L: אורך הכביש.
- N: מספר האוטובוסים הלא רזרביים.
- T: מערך באורך N המתאר את הזמנים שבהם האוטובוסים הלא רזרביים מתוכננים לצאת משדה התעופה.
- W: מערך באורך N המתאר את המהירויות המקסימליות של האוטובוסים הלא רזרביים.
- X: הזמן שלוקח לאוטובוס הרזרבי לנסוע 1 ק"מ.
- M: מספר תחנות המיון.
- S: מערך באורך M המתאר את המרחקים של תחנות המיון משדה התעופה.
- פונקציה זו תיקרא פעם אחת בדיוק לכל טסטקייס, לפני כל הקריאות ל-`arrival_time`.

```
int64 arrival_time(int64 Y)
```

- Y: הזמן שבו האוטובוס הרזרבי (אוטובוס N) מתוכנן לצאת משדה התעופה.
- על פונקציה זו להחזיר את הזמן שבו האוטובוס הרזרבי יגיע למלון.
- פונקציה זו תיקרא בדיוק Q פעמים.

דוגמה

הביטו בסדרת הקריאות הבאה:

```
init(6, 4, [20, 10, 40, 0], [5, 20, 20, 30], 10, 4, [0, 1, 3, 6])
```

אם נתעלם מאוטובוס 4 (שזמן יציאתו עוד לא תוכנן), הטבלה הבאה מראה את הזמני ההגעה הצפויים והאמיתיים של האוטובוסים הלא רזרביים בכל תחנת מיון:

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180

זמני ההגעה לתחנה 0 הם הזמנים שבהם האוטובוסים מתוכננים לצאת משדה התעופה. כלומר, $t_{i,0} = T[i]$ לכל $0 \le i \le 3$.

זמני ההגעה הצפויים והאמיתיים לתחנת המיון 1 מחושבים בצורה הבאה:

- זמני ההגעה הצפויים לתחנה 1:
  - אוטובוס 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - אוטובוס 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - אוטובוס 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.

- ◦ אוטובוס 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- זמני ההגעה האמיתיים לתחנה 1:
  - אוטובוסים 1 ו-3 מגיעים לתחנה 0 לפני אוטובוס 0, אז $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - אוטובוס 3 מגיע לתחנה 0 לפני אוטובוס 1, ולכן $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - אוטובוס 0, אוטובוס 1 ואוטובוס 3 מגיעים לתחנת המיון 0 לפני אוטובוס 2, ולכן $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - אף אוטובוס לא מגיע לתחנה 0 לפני אוטובוס 3, ולכן $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

לאוטובוס 4 לוקח 10 שניות לנסוע 1 ק"מ ועכשיו הוא מתוכנן לצאת משדה התעופה בשנייה ה-0. במקרה זה, הטבלה הבאה מראה את זמני ההגעה של כל אוטובוס. השינוי היחיד בזמני ההגעה הצפויים והאמיתיים של האוטובוסים הלא רזרביים מודגש בקו תחתון.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	$\underline{60}$
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		0		10	10		30	30		60	60

אנחנו רואים שאוטובוס 4 מגיע למלון בשנייה ה-60. לכן, הפונקציה צריכה להחזיר 60.

```
arrival_time(50)
```

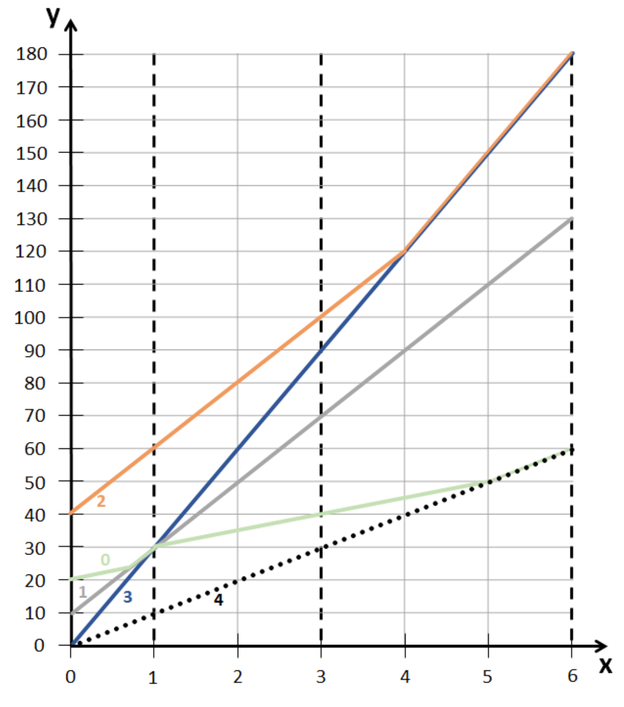
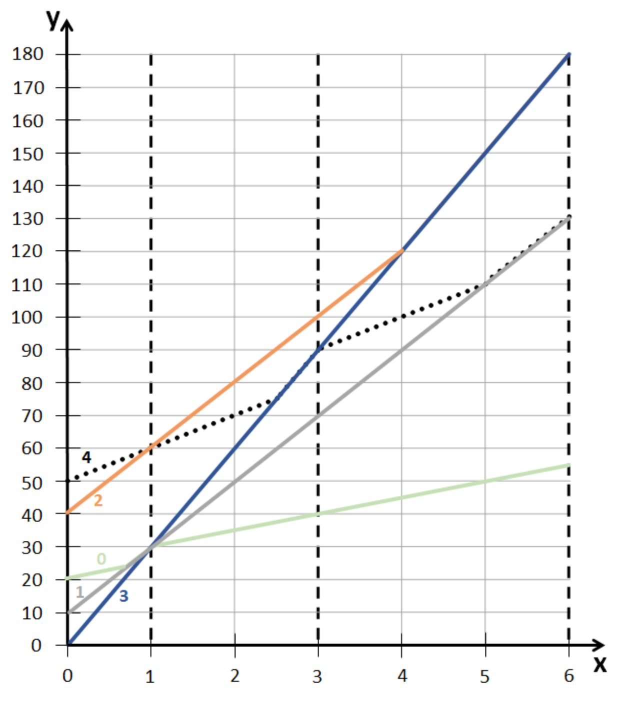
אוטובוס 4 כעת מתוכנן לצאת משדה התעופה בשנייה ה-50. במקרה זה, אין שינויים בזמני ההגעה של האוטובוסים הלא רזרביים בהשוואה לטבלה המקורית. זמני ההגעה מופיעים בטבלה הבאה.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		50		60	60		80	90		120	130

אוטובוס 4 עוקף את אוטובוס 2 שיותר איטי ממנו בתחנת המיון 1, משום שהם מגיעים אליה באותו הזמן. לאחר מכן, אוטובוס 4 מתקבץ עם אוטובוס 3 בין תחנה 1 לתחנה 2, מה שגורם לאוטובוס 4 להגיע לתחנה 2 בשנייה ה-

90 במקום ה-80. אחרי שהוא יוצא מתחנה 2, אוטובוס 4 מתקבץ עם אוטובוס 1 עד שהם מגיעים למלון. אוטובוס 4 מגיע למלון בשנייה ה-130. לכן, על הפונקציה להחזיר 130.

אפשר לסרטט את הזמן שלוקח לכל אוטובוס להגיע לכל מרחק משדה התעופה. ציר ה-x של הגרף מייצג את המרחק משדה התעופה (בקילומטרים), וציר ה-y של הגרף מייצג את הזמן (בשניות). קווים מקווקווים אנכיים מסמנים את מיקומי תחנות המיון. קווים רציפים שונים (מלווים באינדקסים של האוטובוסים) מייצגים את ארבעת האוטובוסים הלא רזרביים. הקו המקווקוו האלכסוני השחור מייצג את האוטובוס הרזרבי.

`arrival_time(0)`	`arrival_time(50)`
	

מגבלות

- $1 \le L \le 10^9$
- $1 \le N \le 1\,000$
- $0 \le T[i] \le 10^{18}$ (לכל i המקיים $0 \le i < N$)
- $1 \le W[i] \le 10^9$ (לכל i המקיים $0 \le i < N$)
- $1 \le X \le 10^9$
- $2 \le M \le 1\,000$
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$
- $1 \le Q \le 10^6$
- $0 \le Y \le 10^{18}$

תתי משימות

1. (9 נקודות) $N = 1, Q \le 1\,000$

2. (10 נקודות) $M = 2, Q \leq 1\,000$
3. (20 נקודות) $N, M, Q \leq 100$
4. (26 נקודות) $Q \leq 5\,000$
5. (35 נקודות) ללא מגבלות נוספות.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1: $L\ N\ X\ M\ Q$
- שורה 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- שורה 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- שורה 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- שורה $5 + k$ $(0 \leq k < Q)$: Y עבור השאילתה k

הגריידר לדוגמה מדפיס את התשובות שלכם בפורמט הבא:

- שורה $1 + k$ $(0 \leq k < Q)$: ערך ההחזרה של `arrival_time` עבור השאילתה k