אפיופניים

אפיופניים הוא שירות שיתוף אופניים שהושק לאחרונה בעיר אפיו. מספר תחנות הוקמו ברחבי העיר, המאפשרות למשתמשים לשאול או להחזיר אופניים בכל תחנה. בזכות הנוחות והעלות הנמוכה, השירות הפך במהירות לאמצעי התחבורה החשוב ביותר בעיר אפיו. עם זאת, אפיופניים נתקלת בבעיה נפוצה בכל שירותי שיתוף האופניים: חוסר איזון בין קצבי ההשאלה וההחזרה בתחנות שונות. כתוצאה מכך, בחלק מהתחנות יש מעט מאוד אופניים, מה שמשאיר את התושבים הסמוכים ללא יכולת לשאול אופניים, בעוד שתחנות אחרות מתמלאות באופניים שהתושבים המקומיים לא צריכים.

כדי לפתור את הבעיה, אפיופניים מתכננת לשלוח משאית איזון מחדש בכל לילה כדי לפזר את האופניים בין התחנות, ולוודא שכל תחנה שומרת על מספר אופניים מתאים. בעיר אפיו ישנן בסך הכל N תחנות, ממוספרות $0, 1, \ldots, N-1$. ממעקב, אפיופניים גילתה שבכל ערב, מספר האופניים בתחנה i הוא תמיד ערך קבוע $A[i]$, ואיש אינו שואל או מחזיר אופניים בלילה. החברה שואפת לכך שיהיו בדיוק $B[i]$ אופניים בתחנה i בכל בוקר.

בין N התחנות הללו, ישנם $N-1$ כבישים ייעודיים למשאית האיזון מחדש. כל כביש מחבר שתי תחנות שונות, והמשאית מוגבלת לנסיעה רק על כבישים אלה. אורכו של כל כביש הוא יחידה אחת. רשת התחנות הזו קשירה, מה שמבטיח שהמשאית יכולה לנסוע בין כל זוג תחנות. בכל לילה, על אפיופניים לשלוח משאית איזון מחדש כדי להבטיח שלכל תחנה i יהיו בדיוק $B[i]$ אופניים. המשאית רשאית להתחיל בכל תחנה ולסיים את מסלולה בכל תחנה.

המשאית ריקה בתחילת האיזון מחדש. בכל פעם שהמשאית נמצאת בתחנה, הנהג רשאי להעמיס כל מספר של אופניים מהתחנה אל המשאית, או לפרוק כל מספר של אופניים מהמשאית אל התחנה. למשאית ולתחנות יש קיבולת בלתי מוגבלת לאופניים, אך מספר האופניים בכל מקום אסור שיירד מתחת לאפס. החברה רוצה לקבוע את מרחק הנסיעה המינימלי האפשרי לסיום האיזון מחדש ואת האסטרטגיה המתאימה.

פרטי המימוש

עליכם לממש את הפונקציה הבאה.

```
std::pair<std::vector<int>, std::vector<long long>>
find_rebalancing_strategy(int N,
                          std::vector<int> A,
                          std::vector<int> B,
                          std::vector<int> U,
                          std::vector<int> V)
```

- A: מערך באורך N, כאשר $A[i]$ הוא מספר האופניים בתחנה i בכל ערב.
- B: מערך באורך N, כאשר $B[i]$ הוא מספר האופניים הרצוי בתחנה i בכל בוקר.

- U, V: מערכים באורך $N-1$ המייצגים את רשת הכבישים. עבור כל $0 \le i < N-1$, הכביש ה-i מחבר את תחנה $U[i]$ ותחנה $V[i]$.
- פונקציה זו נקראת **לכל היותר** $150,000$ **פעמים** עבור כל מקרה בדיקה.

על הפונקציה להחזיר זוג מערכים (X,Y) באותו אורך $k+1$, המייצגים אסטרטגיית איזון מחדש:

- X: רצף האינדקסים של התחנות בהן ביקרה המשאית בסדר כרונולוגי.
- Y: כמות המסירה הנקייה של אופניים בכל תחנה שבוקרה. עבור כל $0 \le j \le k$:
  - אם $Y[j] \ge 0$, הנהג פורק $Y[j]$ אופניים בתחנה $X[j]$ מהמשאית, ומגדיל את ספירת האופניים של התחנה ב-$Y[j]$.
  - אם $Y[j] < 0$, הנהג מעמיס $-Y[j]$ אופניים מתחנה $X[j]$ אל המשאית, ומקטין את ספירת האופניים של התחנה ב-$-Y[j]$.

אסטרטגיית איזון מחדש **תקפה** (X,Y) עם מרחק נסיעה k חייבת לקיים את התנאים הבאים:

- עבור כל $0 \le j \le k$, $0 \le X[j] < N$.
- עבור כל $0 \le j < k$, התחנות $X[j]$ ו-$X[j+1]$ מחוברות ישירות בכביש.
- עבור כל $0 \le j \le k$, $\sum_{t=0}^{j} Y[t] \le 0$.
- עבור כל תחנה $0 \le i < N$ וכל צעד $0 \le j \le k$, יהי $S_{i,j}$ סכום של $Y[t]$ על כל הצעדים t כך ש-$0 \le t \le j$ ו-$X[t] = i$.
  - מספר האופניים בתחנה i לעולם אינו יורד מתחת לאפס: $A[i] + S_{i,j} \ge 0$.
  - בסיום האסטרטגיה, כל תחנה i חייבת להכיל בדיוק $B[i]$ אופניים: $A[i] + S_{i,k} = B[i]$.

אילוצים

- $2 \le N \le 300,000$
- סכום N על כל הקריאות ל-`find_rebalancing_strategy` אינו עולה על $300,000$ בכל מקרה בדיקה.
- $0 \le U[i], V[i] < N$ ו-$U[i] \ne V[i]$ עבור כל i כך ש-$0 \le i < N$.
- $0 \le A[i], B[i] \le 10^9$ עבור כל i כך ש-$0 \le i < N$.
- ניתן לנסוע בין כל זוג תחנות.
- $\sum_{i=0}^{N-1} A[i] = \sum_{i=0}^{N-1} B[i]$.
- קיים לפחות i אחד כך ש-$0 \le i < N$ ו-$A[i] \ne B[i]$.

תתי-משימות וניקוד

כאן, T הוא מספר הקריאות ל-`find_rebalancing_strategy`, ו-$\sum N$ הוא סכום כל ה-N על כל הקריאות ל-`find_rebalancing_strategy`.

תת-משימה	ניקוד	אילוצים נוספים
1	4	התחנות והכבישים מקיימים תכונה P (ראה להלן). קיים בדיוק i אחד כך ש-$0 \le i < N$ ו-$A[i] > 0$.
2	11	$T \le 10$. $N \le 7$. התחנות והכבישים מקיימים תכונה P.
3	16	התחנות והכבישים מקיימים תכונה P.
4	9	קיים בדיוק i אחד כך ש-$0 \le i < N$ ו-$A[i] > 0$.
5	24	$\sum N \le 500$.
6	15	$\sum N \le 5000$.
7	21	ללא אילוצים נוספים.

תכונה P: עבור כל $0 \le i < N - 1$, $U[i] = i$ ו-$V[i] = i + 1$. עבור כל $0 \le i < N$, $A[i] \ne B[i]$.

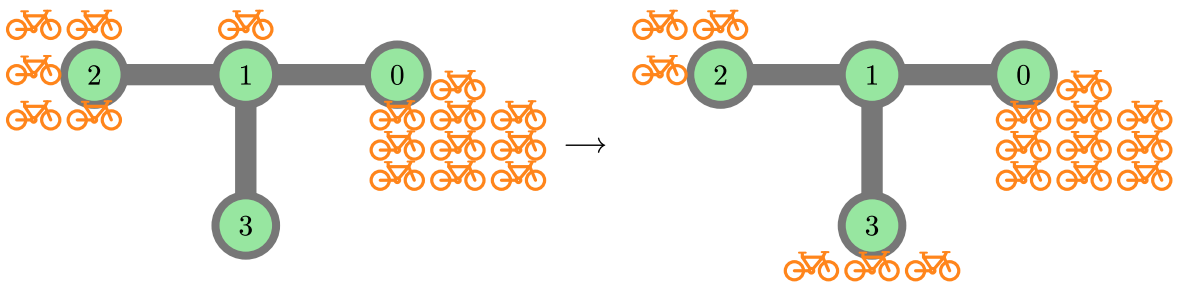
אם מערכי הפלט X ו-Y הם באורך של בדיוק $k^* + 1$, כאשר k^* הוא מרחק הנסיעה המינימלי האפשרי, אך (X, Y) אינה אסטרטגיה תקפה, תקבל 50% מהניקוד.

דוגמאות

דוגמה 1

נניח את הקריאה הבאה:

```
find_rebalancing_strategy(4,
                          [10, 1, 5, 0],
                          [10, 0, 3, 3],
                          [0, 1, 1],
                          [1, 2, 3])
```



ישנן $N = 4$ תחנות בעיר אפיו. בתחילה, לתחנות יש $A = [10, 1, 5, 0]$ אופניים, והמטרה היא שיהיו $B = [10, 0, 3, 3]$ אופניים בכל תחנה.

נניח שמשאית האיזון מחדש מתחילה בתחנה 2, ועוקבת אחר הצעדים הבאים:

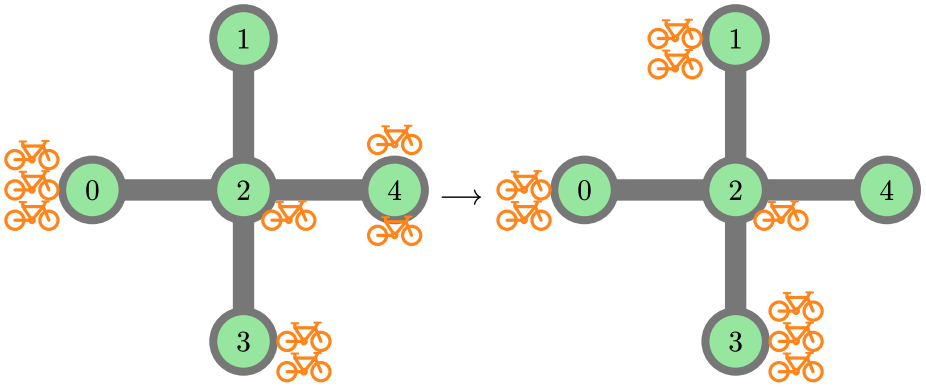
- בתחנה 2, העמס 2 אופניים על המשאית. כעת בתחנה 2 יש 3 אופניים, ובמשאית יש 2 אופניים.
- עבור לתחנה 1 והעמס אופן אחד על המשאית. כעת בתחנה 1 יש 0 אופניים, ובמשאית יש 3 אופניים.
- עבור לתחנה 3, ופרוק 3 אופניים. כעת בתחנה 3 יש 3 אופניים, ובמשאית יש 0 אופניים.

מרחק הנסיעה הכולל הוא 2. מערכי הפלט המתאימים לאסטרטגיה זו הם $X = [2, 1, 3]$ ו-$Y = [-2, -1, 3]$. ניתן להוכיח שאסטרטגיה זו משיגה את המרחק המינימלי. לכן, הפונקציה צריכה להחזיר `([2, 1, 3], [-2, -1, 3])`.

דוגמה 2

נניח את הקריאה הבאה:

```
find_rebalancing_strategy(5,
                          [3, 0, 1, 2, 2],
                          [2, 2, 1, 3, 0],
                          [2, 2, 2, 2],
                          [0, 4, 3, 1])
```



ישנן $N = 5$ תחנות בעיר אפיו. בתחילה, לתחנות יש $A = [3, 0, 1, 2, 2]$ אופניים, והמטרה היא שיהיו $B = [2, 2, 1, 3, 0]$ אופניים בכל תחנה.

נניח שמשאית האיזון מחדש מתחילה בתחנה 0, ועוקבת אחר הצעדים הבאים:

- בתחנה 0, העמס אופן אחד.
- עבור לתחנה 2 והעמס אופן אחד.
- עבור לתחנה 1 ופרוק 2 אופניים.
- עבור לתחנה 2.
- עבור לתחנה 4 והעמס 2 אופניים.
- עבור לתחנה 2 ופרוק אופן אחד.
- עבור לתחנה 3 ופרוק אופן אחד.

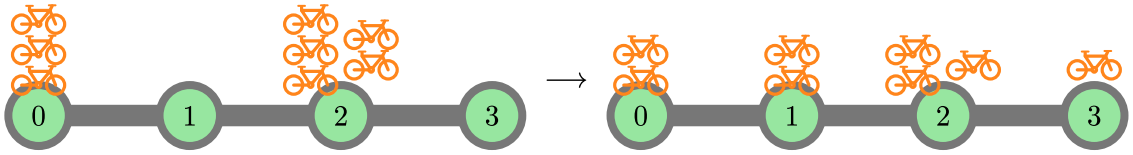
מרחק הנסיעה הכולל הוא 6. מערכי הפלט המתאימים לאסטרטגיה זו הם $X = [0, 2, 1, 2, 4, 2, 3]$ ו-$Y = [-1, -1, 2, 0, -2, 1, 1]$. ניתן להוכיח שאסטרטגיה זו משיגה את המרחק המינימלי. לכן, הפונקציה צריכה להחזיר `([0, 2, 1, 2, 4, 2, 3], [-1, -1, 2, 0, -2, 1, 1])`.

אסטרטגיות תקפות אחרות עם מרחק 6, כגון $X = [0, 2, 3, 2, 4, 2, 1]$ ו-$Y = [-1, 0, 1, 0, -2, 0, 2]$, גם נחשבות נכונות. החזרת מערכי X, Y באורך $7 = 6 + 1$ שאינם אסטרטגיה תקפה, כגון $X = Y = [0, 0, 0, 0, 0, 0, 0]$, תקבל 50% מהניקוד.

דוגמה 3

נניח את הקריאה הבאה:

```
find_rebalancing_strategy(4,
                          [3, 0, 5, 0],
                          [2, 2, 3, 1],
                          [0, 1, 2],
                          [1, 2, 3])
```



פתרון אופטימלי הוא $X = [2, 1, 0, 1, 2, 3]$ ו-$Y = [-1, 1, -1, 1, -1, 1]$. הפונקציה צריכה להחזיר `([2, 1, 0, 1, 2, 3], [-1, 1, -1, 1, -1, 1])`. אסטרטגיות תקפות אחרות, כגון $X = [2, 1, 0, 1, 2, 3]$ ו-$Y = [-2, 2, -1, 0, 0, 1]$, גם נחשבות נכונות.

דוגמה זו מקיימת את האילוצים של תתי-משימות 2 ו-3.

גריידר לדוגמה

השורה הראשונה של הקלט צריכה להכיל מספר שלם בודד T, מספר התרחישים. לאחר מכן יש לתאר את T התרחישים, כל אחד בפורמט המפורט להלן.

פורמט קלט:

```
N
A[0] A[1] ... A[N-1]
B[0] B[1] ... B[N-1]
U[0] V[0]
U[1] V[1]
...
U[N-2] V[N-2]
```

פורמט פלט:

```
k
X[0] X[1] ... X[k]
Y[0] Y[1] ... Y[k]
```