

B. Swapping Cities

Time limit	2 s
Memory limit	512 MB

תיאור

ישנן N ערים באינדונזיה, הממוספרות מ-0 עד $N - 1$. ישנם גם M כבישים דו-סטריים, הממוספרים מ-0 עד $M - 1$. כל כביש מחבר שתי ערים שונות. הכביש i -מחבר את העיר $U[i]$ לעיר $V[i]$ ונסיעה בו מצריכה $W[i]$ יחידות דלק כאשר חוצים אותו במכונית. הערים מחוברות בצורה המאפשרת תנועה בין כל זוג ערים על ידי שימוש בכבישים.

בכל אחד מ- Q הימים הבאים, צמד ערים תהיינה מעוניינות ליצור קשר פוליטי. בפרט, ביום ה- j , העיר $X[j]$ תרצה ליצור קשר פוליטי עם העיר $Y[j]$. על מנת לעשות זאת, על העיר $X[j]$ לשלוח נציג מטעמה לעיר $Y[j]$ במכונית. באופן דומה, גם על העיר $Y[j]$ לשלוח נציג מטעמה לעיר $X[j]$ במכונית.

כדי למנוע גודש, אסור לשתי המכוניות להיפגש בשום זמן. בפרט, אסור לשתי המכוניות להיות באותה העיר באותו הזמן, וגם אסור לשתי המכוניות לנסוע על אותו הכביש בכיוונים נגדיים באותו הזמן. בנוסף, מכונית הנוסעת בכביש חייבת להשלים את הנסיעה בו עד לעיר שבסופו (במילים אחרות, אסור למכונית לבצע פניית פרסה באמצע הנסיעה בכביש). עם זאת, למכוניות מותר לבקר באותה העיר ובאותו הכביש יותר מפעם אחת. יתר על כן, למכוניות גם מותר להמתין בכל עיר, בכל זמן נתון.

מכיוון שמכוניות עם מיכל דלק גדול נחשבות ליקרות, שתי הערים יעדיפו לבחור מסלולים ל-2 המכוניות כך שגודל מיכל הדלק המקסימלי מבין 2 המכוניות יהיה הקטן ביותר האפשרי. יש תחנות דלק בכל עיר עם מלאי אינסופי של דלק, לכן גודל מיכל הדלק הנדרש במכונית יהיה זה המספיק לעבור את הכביש הדורש את כמות הדלק המקסימלית מבין הכבישים אותם עוברת המכונית בדרכה.

משימה

עליכם לממש את הפונקציות: `init` ו-`getMinimumFuelCapacity`.

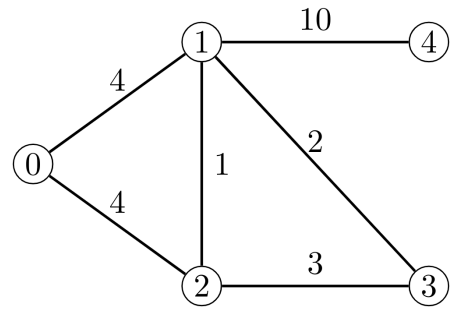
- פונקציה `-(init(N, M, U, V, W))` פונקציה זו תיקרא על ידי הגריידר בדיוק פעם אחת לפני הקריאות ל-`getMinimumFuelCapacity`.
 - N : מספר שלם המייצג את מספר הערים.
 - M : מספר שלם המייצג את מספר הכבישים.
 - U : מערך של M מספרים שלמים המייצג את נקודות הקצה (endpoint) בצד אחד של הכבישים.
 - V : מערך של N מספרים שלמים המייצג את נקודות הקצה (endpoint) בצד השני של הכבישים.
 - W : מערך של M מספרים שלמים המייצג את כמות הדלק המתבזבזת בכל כביש.
- פונקציה `(getMinimumFuelCapacity(X, Y))` פונקציה זו תיקרא על ידי הגריידר בדיוק Q פעמים.
 - X : מספר שלם המייצג את העיר הראשונה.
 - Y : מספר שלם המייצג את העיר השנייה.
- על הפונקציה להחזיר מספר שלם המייצג את גודל מיכל הדלק המקסימלי מבין 2 המכוניות הקטן ביותר האפשרי כך שנציג מטעם העיר X יוכל להגיע לעיר Y ונציג מהעיר Y יוכל להגיע לעיר X בהתאם לחוקים שפורטו בתיאור הבעיה, או -1 אם זה לא אפשרי לעשות זאת.

דוגמאות

בדוגמה הראשונה,

$N = 5, M = 6, U = [0, 0, 1, 1, 1, 2], V = [1, 2, 2, 3, 4, 3], W = [4, 4, 1, 2, 10, 3], Q = 3, X = [1, 2, 0], Y = [2, 4, 1]$.

הדוגמה מומחשת באיור הבא:



הגרייטר תחילה יבצע את הקריאה `init(5, 6, [0, 0, 1, 1, 1, 2], [1, 2, 2, 3, 4, 3], [4, 4, 1, 2, 10], 13)`.

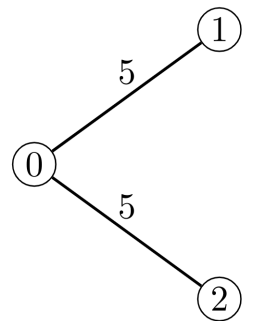
לאחר מכן, הגרייטר יבצע את הקריאות הבאות:

- `getMinimumFuelCapacity(1, 2)`. בהתחלה, המכונת מהעיר הראשונה יכולה לנסוע לעיר השלישית. לאחר מכן, המכונת מהעיר השנייה יכולה לנסוע לעיר הראשונה, והמכונת מהעיר השלישית יכולה לנסוע לעיר השנייה. במצב זה, גודל מיכל הדלק המקסימלי מבין שתי המכונות הוא 3 יחידות של דלק (הדרושות למעבר מהעיר השלישית לעיר השנייה). אין מסלול בו תידרש קיבולת נמוכה יותר של דלק, לכן הפונקציה תחזיר 3.
- `getMinimumFuelCapacity(2, 4)`. כל מכונת הנוסעת מ או אל העיר הרביעית צריכה מיכל המכיל לפחות 10 יחידות דלק, לכן על הפונקציה להחזיר 10.
- `getMinimumFuelCapacity(0, 1)`. על הפונקציה להחזיר 4.

בדוגמה השנייה,

$$N = 3, M = 2, U = [0, 0], V = [1, 2], W = [5, 5], Q = 1, X = [1], Y = [2].$$

הדוגמה מומחשת באיור הבא:



הגרייטר תחילה יבצע את `init(3, 2, [0, 0], [1, 2], [5, 5], 1)`. לאחר מכן, הגרייטר יבצע את הקריאה הבאה:

- `getMinimumFuelCapacity(1, 2)`. בלתי אפשרי למכונת מהעיר הראשונה להגיע לעיר השנייה מבלי לפגוש את המכונת השנייה בדרך, לכן על הפונקציה להחזיר -1.

מגבלות

- $2 \leq N \leq 100\,000$
- $N - 1 \leq M \leq 200\,000$
- $0 \leq U[i] < V[i] < N$
- יש לכל היותר כביש אחד בין כל שתי ערים.
- ניתן לנסוע בין כל שתי ערים על ידי שימוש בכבישים.
- $1 \leq W[i] \leq 10^9$
- $1 \leq Q \leq 200\,000$

$$0 \leq X[j] < Y[j] < N$$

תת משימה 1 (6 נקודות)

כל עיר היא קצה של לכל היותר שני כבישים.

תת משימה 2 (7 נקודות)

$$M = N - 1$$

$$U[i] = 0$$

תת משימה 3 (17 נקודות)

$$Q \leq 5$$

$$N \leq 1000$$

$$M \leq 2000$$

תת משימה 4 (20 נקודות)

$$Q \leq 5$$

תת משימה 5 (23 נקודות)

$$M = N - 1$$

תת משימה 6 (27 נקודות)

ללא מגבלות נוספות.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

```
N M
U[0] V[0] W[0]
U[1] V[1] W[1]
.
.
.
U[M-1] V[M-1] W[M-1]
Q
X[0] Y[0]
X[1] Y[1]
.
.
.
X[Q-1] Y[Q-1]
```

עבור כל קריאה ל- `getMinimumFuelCapacity`, הגריידר מדפיס את הערך החוזר מהפונקציה.