



אלפי האיים

אלפי האיים היא קבוצת איים יפים הנמצאת בים ג'אווה. היא מורכבת מ- N איים, הממוספרים מ-0 עד $N - 1$.

ישנן M סירות קאנו, הממוספרות מ-0 עד $M - 1$, שניתן להשתמש בהן על מנת לשוט בין איים. לכל i המקיים $0 \leq i \leq M - 1$, סירת הקאנו i יכולה לעגון או באי $U[i]$ או באי $V[i]$, וניתן להשתמש בה על מנת לשוט בין איים $U[i]$ ו- $V[i]$. בפרט, כאשר סירת הקאנו עוגנת באי $U[i]$, ניתן לשוט בה מאי $U[i]$ לאי $V[i]$, ולאחר מכן סירת הקאנו תעגון באי $V[i]$. באופן דומה, כשסירת הקאנו עוגנת באי $V[i]$, ניתן לשוט בה מאי $V[i]$ לאי $U[i]$, ולאחר מכן סירת הקאנו תעגון באי $U[i]$. תחילה, סירת הקאנו עוגנת באי $U[i]$. ייתכן שמספר סירות קאנו יכולות לשוט בין אותו זוג האיים. בנוסף ייתכן שמספר סירות קאנו עוגנות באותו האי.

מטעמי בטיחות, סירת קאנו דורשת תחזוקה לאחר כל פעם ששטים בה, מה שאוסר לשוט באותה סירת הקאנו פעמיים ברצף. כלומר, אחרי שיט בסירת קאנו i כלשהי, חייבים לשוט בסירת קאנו אחרת לפני שניתן לשוט בסירת הקאנו i שוב.

בו דנגקלק רוצה לתכנן מסע דרך חלק מהאיים. המסע שלה חוקי אם ורק אם התנאים הבאים מתקיימים.

- היא מתחילה ומסיימת את המסע שלה באי 0.
- היא מבקרת בלפחות אי אחד שאינו אי 0.
- אחרי שהמסע מסתיים, כל סירת קאנו עוגנת באותו אי שהיא עגנה בו לפני המסע. כלומר, סירת הקאנו i , לכל i המקיים $0 \leq i \leq M - 1$, חייבת לעגון באי $U[i]$.

עזרו לבו דנגקלק למצוא מסע חוקי כלשהו ששטים בו לכל היותר 2 000 000 פעמים, או קבעו שלא קיים מסע חוקי כזה. ניתן להוכיח שתחת המגבלות המצוינות במשימה זו (ראו את החלק "מגבלות"), אם קיים מסע חוקי, קיים גם מסע חוקי שלא שטים בו מעל 2 000 000 פעמים.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
union(bool, int[]) find_journey(int N, int M, int[] U, int[] V)
```

- N : מספר האיים.
- M : מספר סירות הקאנו.
- V, U : מערכים באורך M המתארים את סירות הקאנו.
- על פונקציה זו להחזיר או ערך בוליאני או מערך של מספרים שלמים.
 - אם לא קיים מסע חוקי, על הפונקציה להחזיר false.
 - אם קיים מסע חוקי, יש לכם שתי אפשרויות:

- כדי לקבל את הניקוד המלא, על הפונקציה להחזיר מערך של לכל היותר 2 000 000 מספרים שלמים המתאר מסע חוקי. ליתר דיוק, על האיברים במערך זה להיות מספרי סירות הקאנו ששטים בהן במסע (בסדר שבו שטים בהן).
- כדי לקבל ניקוד חלקי, על הפונקציה להחזיר true, מערך של יותר מ-2 000 000 מספרים שלמים, או מערך של מספרים שלמים שלא מתאר מסע חוקי (ראו פירוט ב"תתי משימות").
- פונקציה זו תקרא בדיוק פעם אחת.

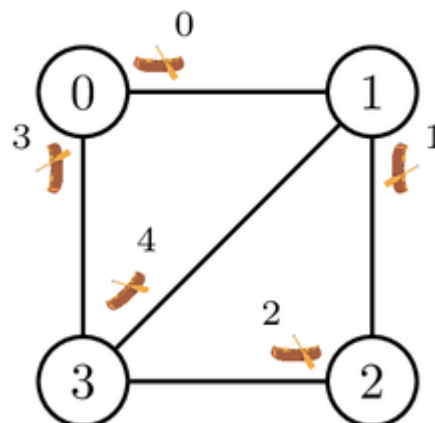
דוגמאות

דוגמה 1

התבוננו בקריאה הבאה:

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

האיים וסירות הקאנו מוצגים באיור מטה.



מסע חוקי אחד הוא כדלהלן. בו דנגקלק ראשית שטה בסירות הקאנו 0, 1, 2, ו-4 בסדר זה. כתוצאה מכך, היא נמצאת באי 1. לאחר מכן, בו דנגקלק יכולה לשוט שוב בסירת הקאנו 0 משום שהיא עוגנת כרגע באי 1 וסירת הקאנו האחרונה שהיא השתמשה בה אינה 0. לאחר ששטה שוב בסירת הקאנו 0, בו דנגקלק נמצאת כעת באי 0. אף על פי כן, סירות הקאנו 1, 2 ו-4 אינן עוגנות באותם האיים שבהם הן עגנו לפני המסע. בו דנגקלק ממשיכה את מסעה על ידי שיט בסירות הקאנו 3, 2, 1, 4, ו-3 שוב. בו דנגקלק חזרה לאי 0 וכל סירות הקאנו עוגנות באותם האיים שבהם עגנו לפני המסע.

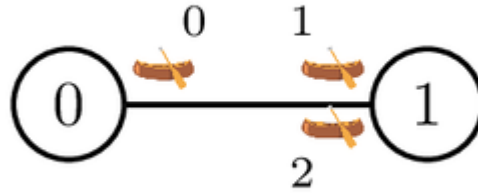
לכן, ערך החזרה [0, 1, 2, 4, 0, 3, 2, 1, 4, 3] מייצג מסע חוקי.

דוגמה 2

התבוננו בקריאה הבאה:

```
find_journey(2, 3, [0, 1, 1], [1, 0, 0])
```

האיים וסירות הקאנו מוצגים באיור מטה.



בו דנגקלק יכולה להתחיל רק בשיט בסירת הקאנו 0, שאחריו היא יכולה לשוט או בסירת הקאנו 1 או ב-2. שימו לב שהיא לא יכולה לשוט בסירת הקאנו 0 פעמיים ברצף. בשני המקרים, בו דנגקלק חוזרת לאי 0. אף על פי כן, סירות הקאנו אינן עוגנות באותם האיים שבהם הן עגנו לפני המסע, ובו דנגקלק לא יכולה לשוט באף סירת קאנו אחר כך, משום שסירת הקאנו היחידה שעוגנת באי 0 היא זו שבה היא בדיוק השתמשה. משום שלא קיים מסע חוקי, על הפונקציה להחזיר false.

מגבלות

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ וגם $0 \leq V[i] \leq N - 1$ (לכל i המקיים $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (לכל i המקיים $0 \leq i \leq M - 1$)

תתי משימות

1. (5 נקודות) $N = 2$
2. (5 נקודות) $N \leq 400$. לכל זוג של איים שונים x ו- y ($0 \leq x < y \leq N - 1$), יש בדיוק שתי סירות קאנו שניתן להשתמש בהן כדי לשוט ביניהם. אחת מהן עוגנת באי x , והשנייה עוגנת באי y .
3. (21 נקודות) $N \leq 1000$, M זוגי, ולכל i זוגי המקיים $0 \leq i \leq M - 1$, סירות הקאנו i ו- $i + 1$ שתיהן יכולות לשוט בין איים $U[i]$ ו- $V[i]$. סירת הקאנו i תחילה עוגנת באי $U[i]$ וסירת הקאנו $i + 1$ תחילה עוגנת באי $V[i]$. פורמלית, $U[i] = V[i + 1]$ וגם $V[i] = U[i + 1]$.
4. (24 נקודות) $N \leq 1000$, M זוגי, ולכל i זוגי המקיים $0 \leq i \leq M - 1$, סירות הקאנו i ו- $i + 1$ שתיהן יכולות לשוט בין איים $U[i]$ ו- $V[i]$. שתי סירות הקאנו תחילה עוגנות באי $U[i]$. פורמלית, $U[i] = U[i + 1]$ וגם $V[i] = V[i + 1]$.
5. (45 נקודות) ללא מגבלות נוספות.

עבור כל טסטקיים שבו קיים מסע חוקי, הפתרון שלכם:

- מקבל את הניקוד המלא אם הוא מחזיר מסע חוקי,

- מקבל 35% מהנקודות אם הוא מחזיר true, מערך של יותר מ-2 000 000 מספרים שלמים, או מערך שלא מתאר מסע חוקי,
- מקבל 0 נקודות אחרת.

עבור כל טסטקייס שבו לא קיים מסע חוקי, הפתרון שלכם:

- מקבל את הניקוד המלא אם הוא מחזיר false,
- מקבל 0 נקודות אחרת.

שימו לב שהניקוד הסופי עבור כל תת משימה הוא הניקוד הקטן ביותר עבור הטסטקייסים בתת המשימה.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1: $N \ M$
- שורה $2 + i$: $U[i] \ V[i] \ (0 \leq i \leq M - 1)$

הגריידר לדוגמה מדפיס את התשובות שלכם בפורמט הבא:

- אם `find_journey` מחזירה `bool`:
 - שורה 1: 0
 - שורה 2: 0 אם `find_journey` מחזירה `false`, או 1 אחרת.
- אם `find_journey` מחזירה `int[]`, נסמן את איברי מערך זה ע"י $c[0], c[1], \dots, c[k-1]$. הגריידר לדוגמה מדפיס:
 - שורה 1: 1
 - שורה 2: k
 - שורה 3: $c[0] \ c[1] \ \dots \ c[k-1]$