משחק תמורות

אליס ובוב הם חברי ילדות, והם אוהבים לשחק משחקי חשיבה. היום, הם משחקים משחק חדש על גרפים.

קופסת המשחק מכילה גרף **קשיר** בעל m צמתים, הממוספרים מ-0 עד $m-1$, ו-e קשתות, הממוספרות מ-0 עד $e-1$. הקשת ה-i מחברת את הצמתים $u[i]$ ו-$v[i]$.

קופסת המשחק מכילה גם תמורה $p[0], p[1], \ldots, p[n-1]$ באורך n, כאשר $m \leq n$. תמורה היא מערך שבו כל מספר מ-0 עד $n-1$ מופיע בדיוק פעם אחת, בסדר כלשהו. ה**ניקוד** של תמורה p הוא מספר האינדקסים i כך ש-$p[i] = i$.

המשחק יימשך לכל היותר 10^{100} תורות. בכל תור, הדברים הבאים מתרחשים:

1. אם אליס מחליטה לסיים את המשחק, המשחק נעצר.
2. אחרת, אליס בוחרת **אינדקסים שונים** $t[0], t[1], \ldots, t[m-1]$, כאשר $0 \leq t[i] < n$. שימו לב שהמשחק **לא** דורש שיתקיים $t[0] < t[1] < \ldots < t[m-1]$.
3. בוב בוחר אינדקס $0 \leq j < e$ מבין הקשתות בגרף ומחליף את $p[t[u[j]]]$ ואת $p[t[v[j]]]$.

אליס מעוניינת למקסם את הניקוד הסופי של התמורה, בעוד בוב מעוניין למזער את הניקוד הסופי של התמורה.

משימתכם היא לעזור לאליס לשחק נגד בוב, כאשר המהלכים של בוב מסומלצים על ידי גריידר.

נגדיר את *הניקוד האופטימלי* כניקוד הסופי של התמורה אם גם אליס וגם בוב משחקים באופן אופטימלי.

עליכם לקבוע את הניקוד האופטימלי של התמורה ואז לשחק את המשחק נגד בוב כדי להשיג **לפחות** את הניקוד האופטימלי הזה כעבור מספר תורות.

שימו לב שהאסטרטגיה של אליס צריכה לעבוד ללא תלות במהלכים של בוב, גם אם בוב יעשה מהלכים לא אופטימליים.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
int Alice(int m, int e, std::vector<int> u, std::vector<int> v,
          int n, std::vector<int> p)
```

- m: מספר הצמתים בגרף.
- e: מספר הקשתות בגרף.
- u ו-v: מערכים באורך e המתארים את הקשתות שבגרף.
- n: אורך התמורה.

- p: מערך באורך n המתאר את התמורה.
- פונקציה זו תיקרא בדיוק פעם אחת.
- על פונקציה זו צריכה להחזיר מספר שלם – הניקוד האופטימלי של המשחק.

מתוך פונקציה זו, אתם יכולים לקרוא לפונקציה הבאה:

```
int Bob(std::vector<int> t)
```

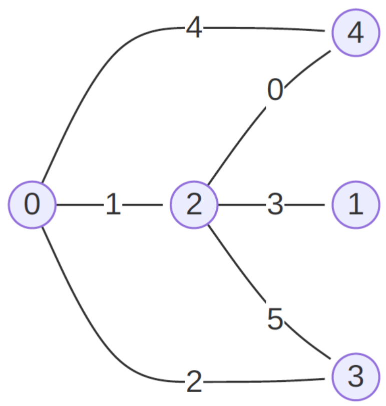
- t: מערך באורך m, המכיל אינדקסים יחודיים, כאשר $0 \leq t[i] < n$ וגם $t[i] \neq t[j]$ לכל $i \neq j$.
- פונקציה זו מחזירה מספר שלם יחיד j המקיים $0 \leq j < m$.
- פונקציה זו יכולה להיקרא מספר פעמים.

דוגמה

התבוננו בקריאה הבאה:

```
Alice(5, 6, [4, 0, 3, 1, 4, 2], [2, 2, 0, 2, 0, 3],
      10, [8, 2, 7, 6, 1, 5, 0, 9, 3, 4])
```

הגרף הוא כדלקמן:



ו-p היא בהתחלה $[8,2,7,6,1,5,0,9,3,4]$.

בהינתן האילוצים האלו, ניתן להוכיח כי הניקוד האופטימלי של התמורה הוא 1.

נניח כי אליס מבצעת את 4 המהלכים הבאים:

הארגומנט t ל-Bob	ערך החזרה של Bob	האינדקסים המתאימים ב-p	p לאחר ההחלפה של בוב
$[3,1,5,2,0]$	5	$5,2$	$[8,2,5,6,1,7,0,9,3,4]$
$[9,3,7,2,1]$	0	$1,7$	$[8,9,5,6,1,7,0,2,3,4]$
$[5,6,7,8,9]$	1	$5,7$	$[8,9,5,6,1,2,0,7,3,4]$
$[7,5,2,3,6]$	3	$5,2$	$[8,9,2,6,1,5,0,7,3,4]$

שימו לב שאליס ובוב לא בהכרח משחקים את המהלכים האופטימליים. מהלכים אלו מוצגים להמחשה בלבד. שימו גם לב לכך שאליס יכלה לסיים את המשחק מיידית, שכן הניקוד ההתחלתי של התמורה הוא כבר 1.

אחרי שאליס ביצעה את כל המהלכים לעיל, הניקוד העדכני של התמורה הוא 3 ($p[2] = 2, p[5] = 5, p[7] = 7$).

לבסוף, הפונקציה `Alice()` תחזיר 1 – הניקוד האופטימלי של התמורה.

שימו לב כי אף על פי שאליס השיגה ניקוד של 3 כששיחקה נגד בוב, אתם תקבלו 0 נקודות אם ערך החזרה של `Alice()` יהיה 3 במקום 1.

אילוצים

- $2 \leq m \leq 400$
- $m - 1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- הגרף קשיר, ולא מכיל לולאות עצמיות או קשתות כפולות.
- p היא תמורה, כלומר $p[i] \neq p[j]$ לכל $i \neq j$.

תת משימות

1. (6 נקודות) $m = 2$
2. (6 נקודות) $e > m$
3. (10 נקודות) $e = m - 1$
4. (24 נקודות) $e = m = 3$
5. (24 נקודות) $e = m = 4$
6. (30 נקודות) $e = m$

בכל תת משימה, אתם יכולים לקבל ניקוד חלקי. יהי r היחס המירבי של $\frac{k}{n}$ על פני כל הטסטים של תת משימה, כאשר k הוא מספר התורות (כלומר, קריאות ל- `Bob()`). אז הניקוד שלכם בתת המשימה מוכפל במספר הבא:

תנאי	כופל
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r - 2)$
$r \leq 3$	1

בפרט, אם תפתרו את הבעיה תוך לכל היותר $3n$ תורות, תקבלו ניקוד מלא עבור תת המשימה. שימוש ביותר מ-$12n$ תורות יגרום לקבלת 0 נקודות בתת המשימה (מוצג כ-`Output isn't correct`).

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1: $m\ e$
- שורה $2+i$ ($0 \le i \le e-1$): $u[i]\ v[i]$
- שורה $2+e$: n
- שורה $3+e$: $p[0]\ p[1]\ \ldots\ p[n-1]$

הגריידר לדוגמה מדפיס את הפלט בפורמט הבא:

- שורה 1: התמורה הסופית p.
- שורה 2: ערך החזרה של `Alice()`.
- שורה 3: הניקוד של התמורה הסופית.
- שורה 4: מספר התורות.