



## טיול ארוך ביותר

מארגני IOI 2023 בצרות גדולות! הם שכחו לתכנן את הטיול לאופוסטסר ליום הקרוב. אבל אולי עוד לא מאוחר מדי...

ישנם  $N$  ציוני דרך באופוסטסר הממוספרים מ-0 עד  $N - 1$ . חלק מזוגות ציוני הדרך האלה מחוברים על ידי **כבישים דו-סטריים**. כל זוג ציוני דרך מחובר על ידי כביש אחד לכל היותר. המארגנים אינם יודעים אילו ציוני דרך מחוברים על ידי כבישים.

נאמר שהצפיפות של מערכת הכבישים של אופוסטסר היא **לפחות**  $\delta$  אם לכל 3 ציוני דרך שונים, יש לפחות  $\delta$  כבישים ביניהם. במילים אחרות, לכל שלשת ציוני דרך  $(u, v, w)$  כך ש- $0 \leq u < v < w < N$ , מבין זוגות ציוני הדרך  $(u, v)$ ,  $(v, w)$  ו- $(u, w)$ , לפחות  $\delta$  זוגות מחוברים על ידי כביש.

ידוע למארגנים מספר שלם חיובי  $D$  כך שהצפיפות של מערכת הכבישים היא לפחות  $D$ . שימו לב שערכו של  $D$  לא יכול להיות גדול מ-3.

המארגנים יכולים לעשות שיחות למוקדן באופוסטסר כדי לאסוף מידע לגבי חיבורי הכבישים בין נקודות ציון מסוימות. בכל שיחה, הם חייבים לתת שני מערכים לא ריקים של נקודות ציון  $[A[0], \dots, A[P - 1]]$  ו- $[B[0], \dots, B[R - 1]]$ . נקודות הציון שניתנות צריכות להיות שונות אחת מהשנייה, כלומר,

- $A[i] \neq A[j]$  לכל  $i$  ו- $j$  המקיימים  $0 \leq i < j < P$ ;
- $B[i] \neq B[j]$  לכל  $i$  ו- $j$  המקיימים  $0 \leq i < j < R$ ;
- $A[i] \neq B[j]$  לכל  $i$  ו- $j$  המקיימים  $0 \leq i < P$  וגם  $0 \leq j < R$ .

עבור כל שיחה, המוקדן מדווח האם ישנו כביש שמחבר נקודת ציון מ- $A$  ונקודת ציון מ- $B$ . ליתר דיוק, המוקדן עובר על כל הזוגות  $i$  ו- $j$  המקיימים  $0 \leq i < P$  ו- $0 \leq j < R$ . אם, עבור אחד מהם, נקודות הציון  $A[i]$  ו- $B[j]$  מחוברות על ידי כביש, המוקדן מחזיר true. אחרת, המוקדן מחזיר false.

**טיול** מאורך  $l$  הוא רצף של נקודות ציון **שונות**  $t[0], t[1], \dots, t[l - 1]$ , שבו לכל  $i$  בין 0 ל- $l - 2$ , כולל, נקודת הציון  $t[i]$  ונקודת הציון  $t[i + 1]$  מחוברות על ידי כביש. טיול מאורך  $l$  נקרא **טיול ארוך ביותר** אם לא קיים אף טיול באורך לפחות  $l + 1$ .

משימתכם היא לעזור למארגנים למצוא טיול ארוך ביותר באופוסטסר בעזרת ביצוע שיחות למוקדן.

## פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
int[] longest_trip(int N, int D)
```

- $N$ : מספר נקודות הציון באופוסטטר.
- $D$ : הצפיפות המינימלית המובטחת של מערכת הכבישים.
- על פונקציה זו להחזיר מערך  $t = [t[0], t[1], \dots, t[l-1]]$ , המייצג טיול ארוך ביותר.
- פונקציה זו יכולה להיקרא **מספר פעמים** בכל טסטקייס.

הפונקציה לעיל יכולה לבצע קריאות לפונקציה הבאה:

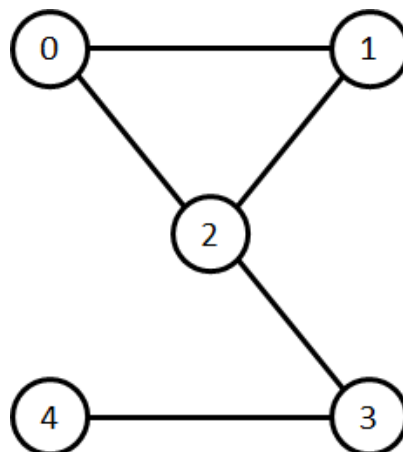
```
bool are_connected(int[] A, int[] B)
```

- $A$ : מערך לא ריק של נקודות ציון שונות.
  - $B$ : מערך לא ריק של נקודות ציון שונות.
  - $A$  ו- $B$  צריכים להיות זרים.
  - פונקציה זו מחזירה true אם קיימת נקודת ציון מ- $A$  ונקודת ציון מ- $B$  המחוברות על ידי כביש. אחרת, היא מחזירה false.
  - פונקציה זו יכולה להיקרא לכל היותר 32 640 פעמים בכל הרצה של longest\_trip, ולכל היותר 150 000 פעמים בסך הכל.
  - האורך הכולל של המערכים  $A$  ו- $B$  המועברים לפונקציה זו בכל ההרצות שלה לא יכול לחרוג מ-1 500 000.
- הגריידר אינו אדפטיבי. כל הגשה נבדקת על אותה קבוצה של טסטקייסים. כלומר, הערכים של  $N$  ושל  $D$ , בנוסף לזוגות נקודות הציון המחוברות על ידי כבישים, קבועים עבור כל קריאה ל-longest\_trip בתוך כל טסטקייס.

## דוגמאות

### דוגמה 1

התבוננו בתרחיש שבו  $D = 1$ ,  $N = 5$ , וחיבורי הכבישים הם כפי שמתואר בתרשים הבא:



הפונקציה longest\_trip נקראת באופן הבא:

```
longest_trip(5, 1)
```

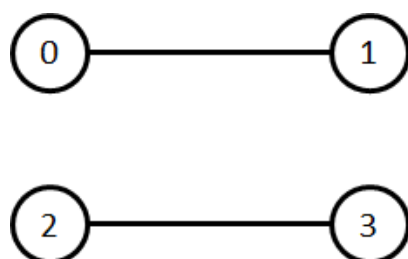
הפונקציה יכולה לבצע קריאות ל-are\_connected כדלקמן:

קריאה	זוגות מחוברים על ידי כביש	ערך החזרה
<code>are_connected([0], [1, 2, 4, 3])</code>	(0, 1) ו- (0, 2)	true
<code>are_connected([2], [0])</code>	(2, 0)	true
<code>are_connected([2], [3])</code>	(2, 3)	true
<code>are_connected([1, 0], [4, 3])</code>	אף אחד	false

אחרי הקריאה הרביעית, מסתבר שאף אחד מהזוגות (1, 4), (0, 4), (1, 3) ו- (0, 3) לא מחובר על ידי כביש. משום שהצפיפות של מערכת הכבישים היא לפחות  $D = 1$ , אפשר לראות שמבין השלשה (0, 3, 4), הזוג (3, 4) חייב להיות מחובר על ידי כביש. באופן דומה, נקודות הציון 0 ו-1 חייבות להיות מחוברות.

בנקודה זו, אפשר להסיק ש- $t = [1, 0, 2, 3, 4]$  הוא טיול מאורך 5, ושלא קיים טיול מאורך גדול מ-5. לכן, הפונקציה `longest_trip` יכולה להחזיר `[1, 0, 2, 3, 4]`.

התבוננו בתרחיש אחר שבו  $N = 4$ ,  $D = 1$ , והכבישים בין נקודות הציון הם כפי שמתואר בתרשים הבא:



הפונקציה `longest_trip` נקראת בצורה הבאה:

```
longest_trip(4, 1)
```

בתרחיש הזה, אורך הטיול הארוך ביותר הוא 2. לכן, אחרי כמה קריאות לפונקציה `are_connected`, הפונקציה `longest_trip` יכולה להחזיר אחד מבין `[0, 1]`, `[1, 0]`, `[2, 3]` או `[3, 2]`.

דוגמה 2

תת-משימה 0 מכילה טסטקייס נוסף לדוגמה עם  $N = 256$  נקודות ציון. הטסטקייס הזה כלול בחבילת הקבצים המצורפת שאתם יכולים להוריד ממערכת התחרות.

## מגבלות

- $3 \leq N \leq 256$
- סכום ערכי  $N$  בכל הקריאות ל-`longest_trip` לא חורג מ-1024 בכל טסטקייס.
- $1 \leq D \leq 3$

## תתי משימות

1. (5 נקודות)  $D = 3$
2. (10 נקודות)  $D = 2$
3. (25 נקודות)  $D = 1$ . נסמן ב- $l^*$  את אורך הטיול הארוך ביותר. הפונקציה `longest_trip` לא חייבת להחזיר טיול באורך  $l^*$ . במקום זאת, עליה להחזיר טיול באורך לפחות  $\left\lceil \frac{l^*}{2} \right\rceil$ .
4. (60 נקודות)  $D = 1$

בתת משימה 4, הניקוד שלכם יקבע על פי מספר הקריאות לפונקציה `are_connected` עבור הרצה אחת של `longest_trip`. נסמן ב- $q$  את מספר הקריאות המקסימלי מבין כל ההרצות של `longest_trip` על פני כל הטסטקייסים בתת המשימה. הניקוד שלכם בתת המשימה הזו מחושב לפי הטבלה הבאה:

ניקוד	תנאי
20	$2\,750 < q \leq 32\,640$
30	$550 < q \leq 2\,750$
45	$400 < q \leq 550$
60	$q \leq 400$

אם בלפחות אחד מהטסטקייסים, הקריאות לפונקציה `are_connected` לא עונות על המגבלות המתוארות בפרטי מימוש, או שהמערך שמוחזר על ידי `longest_trip` אינו נכון, הניקוד של פתרוכם על תת המשימה הזו יהיה 0.

## גריידר לדוגמה

נסמן ב- $C$  את מספר התרחישים, כלומר, מספר הקריאות ל-`longest_trip`. הגריידר לדוגמה קורא את הקלט בפורמט הבא:

• שורה 1:  $C$

התיאורים של  $C$  תרחישים לאחר מכן.

הגריידר לדוגמה קורא את התיאור של כל תרחיש בפורמט הבא:

• שורה 1:  $N \ D$

• שורה  $i + 1$ :  $U_i[0] \ U_i[1] \ \dots \ U_i[i - 1]$  ( $1 \leq i < N$ )

כאן, כל  $U_i$  ( $1 \leq i < N$ ) הוא מערך באורך  $i$ , המתאר אילו זוגות של נקודות ציון מחוברים על ידי כביש. לכל  $i$  ו- $j$  כך ש- $1 \leq i < N$  וגם  $0 \leq j < i$ :

- אם נקודות ציון  $i$  ו- $j$  מחוברות על ידי כביש, הערך של  $U_i[j]$  צריך להיות 1;
- אם אין כביש המחבר בין נקודות ציון  $i$  ו- $j$ , אז הערך של  $U_i[j]$  צריך להיות 0.

בכל תרחיש, לפני הקריאה ל-`longest_trip`, הגריידר לדוגמה בודק האם הצפיפות של מערכת הכבישים היא לפחות  $D$ . אם תנאי זה לא מתקיים, הוא מדפיס את ההודעה `Insufficient Density` ומסיים את ריצתו.

אם הגריידר לדוגמה מזהה הפרת פרוטוקול, הפלט של הגריידר לדוגמה הוא Protocol Violation:  $\langle \text{MSG} \rangle$  כאשר  $\langle \text{MSG} \rangle$  הוא אחת מהודעות השגיאה הבאות:

- `invalid_array`: בקריאה ל-`are_connected`, לפחות אחד מבין המערכים  $A$  ו- $B$ 
  - ריק, או
  - מכיל איבר שאינו מספר שלם בין 0 ל- $N - 1$  כולל, או
  - מכיל את אותו האיבר לפחות פעמיים
- `non-disjoint arrays`: בקריאה ל-`are_connected`, המערכים  $A$  ו- $B$  לא זרים.
- `too many calls`: מספר הקריאות שנעשו ל-`are_connected` חורג מ-32 640 בהרצה הנוכחית של `longest_trip`, או חורג מ-150 000 בסך הכל.
- `too many elements`: מספר נקודות הציון שהעוברו ל-`are_connected` לאורך כל הקריאות חורג מ-1 500 000.

אחרת, נסמן את איברי המערך שהוחזרו על ידי `longest_trip` בתרחיש על ידי  $t[0], t[1], \dots, t[l-1]$  עבור  $l$  אי שלילי כלשהו. הגריידר לדוגמה מדפיס שלוש שורות לתרחיש זה בפורמט הבא:

- שורה 1:  $l$
- שורה 2:  $t[0] \ t[1] \ \dots \ t[l-1]$
- שורה 3: מספר הקריאות ל-`are_connected` בתרחיש זה.

לסיום הגריידר לדוגמה מדפיס:

- שורה  $1 + 3 \cdot C$ : המספר המקסימלי של קריאות ל-`are_connected` לאורך כל הקריאות ל-`longest_trip`