

## אליס, בוב וגל המעגל

מוסד המעגלים של סייברלנד מונה  $n$  חברים. לכל חבריה יש את המספר האהוב עליהם ושם ייחודי (המספר האהוב עשוי לא להיות ייחודי).

$m$  מכתבים נשלחו בין החברים. לכל מכתב יש שולח ומקבל, והתוכן של המכתב הוא המספר האהוב על השולח.

כל החברים מחשבים את הסכום של התכנים (המספרים האהובים של השולחים) שהם מקבלים ולוקחים אותו מודולו  $65536$  ( $2^{16}$ ) בתור מספר התוצאה שלהם.

המשימה שלכם היא לקבוע את כל מספרי התוצאות.

אולם, המצב הוא לא ישיר כפי שהוא נראה. אליס, בוב וגל המעגל החליטו לפתור את הבעיה בדרך קצת יותר מסובכת:

- אליס מכירה את כל  $n$  החברים (שם ומספר אהוב) אבל אין לה שום מידע על המכתבים. היא צריכה לשלוח מחרוזת בינארית לגל באורך לכל היותר  $10^5$ .
- בוב מכיר את כל  $m$  המכתבים (שם השולח והמקבל), אבל אין לו שם מידע על החברים. הוא צריך לשלוח מחרוזת בינארית לגל באורך לכל היותר  $10^5$ .
- גל המעגל יכול לקבל מחרוזות בינאריות שהתקבלו מאליס ומבוב וליצור מחרוזת בינארית אשר מורכבת מ- $16m$  ביטים בתור פלט. אך בגלל כוח המחשוב המוגבל שלו, גל יכול לבצע רק פעולות לוגיות בסיסיות (למשל AND, XOR, NOT).

בקטע הבא, נציג איך המעגל עובד בפירוט.

### פרטי המעגל

השער הוא האלמנט הבסיסי של מעגל. שער מכיל אפס או שני קלטים בוליאניים (תלוי בסוג השער) ופלט בוליאני אחד. יש שני סוגי שערים: שערי קלט ושערי חישוב.

- לשערי קלט אין קלט והם מייצגים את הביטים מהמחרוזות הבינאריות שנשלחו מאליס ומבוב.
  - יהיו  $l_A + l_B$  שערי קלט, ממוספרים מ-0 עד  $(l_A + l_B - 1)$ , כאשר  $l_A, l_B$  הם האורכים של המחרוזות שנשלחו מאליס ובוב בהתאמה.
  - עבור  $0 \leq i < l_A$ , הפלט של השער ה- $i$  הוא הביט ה- $i$  של המחרוזת מאליס.
  - עבור  $0 \leq i < l_B$ , הפלט של השער ה- $(i + l_A)$  הוא הביט ה- $i$  של המחרוזת מבוב.

- לשערי חישוב יש שני קלטים והם מייצגים את תהליך החישוב.

- המספרים של שערי החישוב מתחילים מ- $(l_A + l_B)$ .
- לכל שער חישוב, עליכם לספק מספר של שני שערי תלות בתור קלט, וסוג הפעולה ( $0 \leq p \leq 15$ ).

- כדי למנוע תלות מעגלית, המספרים של שני שערי התלות חייבים להיות קטנים מהמספר של שער החישוב.

- אם הפלטים של שני שערי התלות הם  $x_0, x_1$  בהתאמה  $\{0, 1\}$ , אז הפלט של שער החישוב הוא:

$$f(p, x_0, x_1) = \left\lfloor \frac{p}{2^{x_0+2x_1}} \right\rfloor \bmod 2$$

להלן מספר דוגמאות שעשויות להיות שימושיות:

NOT $x_0$ $f(5, x_0, x_1)$	$x_1$ XOR $x_0$ $f(6, x_0, x_1)$	$x_1$ OR $x_0$ $f(14, x_0, x_1)$	$x_1$ AND $x_0$ $f(8, x_0, x_1)$	$x_1$	$x_0$
1	0	0	0	0	0
0	1	1	0	0	1
1	1	1	0	1	0
0	0	1	1	1	1

### פרטי מימוש

שימו לב:

- אינדקסים של מערך יתחילו מ-0. למשל, אם  $a$  מערך בגודל  $n$  אז  $a[0]$  עד  $a[n-1]$  - מידע חוקי. גישה לאינדקסים מעבר לטווח זה עלול לגרום out-of-bounds error.
- כל המחרוזות מסתיימות בתו  $\backslash 0$  null.

אליס

```
int alice(const int n, const char names[][5], const unsigned short numbers[], bool outputs_alice[]);
```

כיוון	ערך	אורך	משמעות	מגבלות
קלט	n	1	$n$	$0 \leq n \leq 700$
	names	$n$	השם של כל חברה	שמות הם ייחודיים, מכילים אותיות קטנות באנגלית בלבד, ובעלי אורך מקסימלי של 4 תווים.
	numbers	$n$	המספר האהוב על כל חברה.	כל מספר בטווח בין 0 ל-65535.
פלט	outputs_alice	$l_A$	המחרוזת הבינארית שנשלחת לגל.	
	ערך החזרה	1	$l_A$	עליכם לוודא ש- $l_A$ לא עובר את $10^5$ וכאשר $n$ זהה, $l_A$ חייב להיות קבוע.

בוב

```
int bob(const int m, const char senders[][5], const char recipients[][5], bool outputs_bob[]);
```

כיוון	ערך	אורך	משמעות	מגבלה
קלט	m	1	$m$	$0 \leq m \leq 1000$
	senders	$m$	שם השולח על כל מכתב.	השמות שמופיעים בקלט של אליס.
	recipients	$m$	השם של המקבל על כל מכתב.	
פלט	outputs_bob	$l_B$	המחרוזת הבינארית שנשלחת לגל.	
	ערך החזרה	1	$l_B$	עליכם לוודא ש- $l_B$ לא עובר את $10^5$ וכאשר $m$ זהה, $l_B$ חייב להיות קבוע.

גל המעגל

כדי להבטיח שתהליך החישוב של גל הוא כמו מעגל כללי, אתם לא יכולים להשיג באופן ישיר את המחרוזות הבינאריות שאליס ובוב שלחו לגל. אתם רק יודעים את האזכורים של שתי המחרוזות האלו וצריכים לפלוט את מבנה המעגל.

```
int circuit(const int la, const int lb, int operations[], int operands[][2], int outputs_circuit[][16]);
```

כיוון	ערך	אורך	משמעות	מגבלה
קלט	la	1	$l_A$	
	lb	1	$l_B$	
פלט	operations	$l$	סוג הפעולה שבוצעה על ידי כל שער במעגל.	מספר בין 0 ל-15.
	operands	$l$	האופרנד ששומש על ידי כל שער במעגל.	המספר חייב להיות קטן מהתוויות של השער הנוכחי.
	outputs_circuit	$n$	תווית השער של פלט המעגל.	$outputs\_circuit[i][j]$ מסמן את הביט ה- $j$ (בספירה מהביט הכי פחות משמעותי) בתוצאה הסופית של החברה ה- $i$ . כל החברים מסודרים לפי הקלט של אליס.
	ערך החזרה	1	$l$ , אשר מייצג את מספר השערים הכולל (כולל שערי הקלט).	עליכם לוודא שמתקיים $l \leq 2 \times 10^7$

למרות שאתם יכולים לשנות את המידע של שערים עם אינדקסים קטנים מ- $l_A + l_B$  במערכים operations ו-operands, הגריידר יתעלם משינויים כאלו.

דוגמה

```
alice(3, {"alic", "bob", "circ"}, {10000, 20000, 30000}, outputs_alice);
bob(5, {"alic", "bob", "bob", "circ", "circ"}, {"circ", "circ", "alic", "circ", "circ"}, outputs_bob);
```

קריאה זו מייצגת את התרחיש הבא:

- אליס יודעת שיש 3 חברים, לחברה בשם alic- המספר המועדף הוא 10000 וכל פלט אפשרי של `alice()` הוא,
  - ערך ההחזרה של `alice()` הוא 2, אשר מייצג  $l_A = 2$ .
  - בתוך הפונקציה `alice()`, לקבוע `outputs_alice[0] = 1, outputs_alice[1] = 0`, אשר מייצגים שהמחרוזת הבינארית שהיא התוצאה היא 10.
- בוב יודע שיש 5 מכתבים, המכתב הראשון הוא בין alic ל-circ וכל פלט אפשרי של `bob()` הוא,
  - ערך ההחזרה של `bob()` הוא 3, שמייצג  $l_B = 3$ .
  - בתוך הפונקציה `bob()`, לקבוע `outputs_bob[0] = 1, outputs_bob[1] = 1, outputs_bob[2] = 0`, אשר מייצגים שהמחרוזת הבינארית שהיא התוצאה היא 110.

בהתבסס על הפלטים הקודמים של `alice()` ו-`bob()`, תהיה הקריאה הבאה:

```
circuit(2, 3, operations, operands, outputs_circuit);
```

פלט נכון עבור הפונקציה יהיה:

- ערך ההחזרה של `circuit()` הוא 7, שאומר שנוסיף שני שערי חישוב, ממוספרים 5 ו-6.
- בתוך `circuit()`, להגדיר את `operations`, `operands` ו-`outputs_circuit` באופן הבא:
  - `operations = {-1, -1, -1, -1, -1, 8, 14}` כאשר אנחנו משתמשים ב-1- כדי לסמן מידע ששערי הקלט מתעלמים ממנו;
  - `operands = {{-1, -1}, {-1, -1}, {-1, -1}, {-1, -1}, {-1, -1}, {0, 4}, {2, 5}}`
  - `outputs_circuit = {{5, 5, 5, 5, 5, 6, 5, 5, 5, 6, 6, 6, 5, 5, 6, 5}, ...}` המערך הזה די ארוך. ניתן לבדוק בקובץ `abc.cpp` בקבצים המצורפים כדי לראות את המערך המלא.

בהינתן הפלט תהליך החישוב הוא,

- הוסף שער חישוב מסוג 8 עם קלט משער 0 ומשער 4. הפלט של שער 0 הביט 0-ה של המחרוזת של אליס, שהוא 1; הפלט של שער 4 הוא הביט 2-ה של המחרוזת של בוב, שהוא 0. לכן הפלט של שער 5 הוא  $f(8, 0, 1) = 0 \text{ AND } 1 = 0$ .
- הוסף שער חישוב מסוג 14 עם קלט משער 2 ומשער 5. הפלט של שער 2 הוא הביט 0-ה של המחרוזת של בוב שהוא 1; הפלט של שער 5 הוא 0. אז הפלט של שער 6 הוא  $f(14, 1, 0) = 1 \text{ OR } 0 = 1$ .
- `output_circuit[0]` מייצג את התוצאה הסופית של alic, שהיא  $(0100111000100000)_2 = 20000$ . בגלל ש-alic מקבלת מכתב רק מ-bob, התוצאה הסופית של alic היא 20000.
- התוצאה הסופית של bob צריכה להיות 0, בגלל שהוא לא מקבל שום מכתב; התוצאה הסופית של circ צריכה להיות  $(10000 + 20000 + 30000 + 30000) \bmod 65536 = 24464$ .

`abc.cpp` המצורף יכול לעבור את הדוגמה הזו, אבל אנו לא מבטיחים שהוא יכול לעבור טסטקייסים אחרים.

## מגבלות

לכל הטסטקייסים:

- $0 \leq n \leq 700, 0 \leq m \leq 1000$ .
- כל השמות שונים, מורכבים מאותיות אנגליות קטנות בלבד, ומאורך לכל היותר 4 תווים.
- המספר האהוב על כל חבר הוא בין 0 ל-65535.
- השמות של כל השולחים והמקבלים מופיעים במערך הקלט של אליס `names`.
- ל-`alice()` ול-`bob()` יש מגבלת זיכרון של MiB 2048 ומגבלת זמן של 0.02 שניות, בהתאמה.
- ל-`circuit()` יש מגבלת זכרון של MiB 2048 ומגבלת זמן של 7 שניות.

בהרצה הסופית, ל-`alice()` ו-`bob()` עלולים להיקרא מספר פעמים בטסטקייס יחיד. מגבלת הזמן של 0.02 שניות היא לכל קריאה.

## תתי משימות

תת משימה מסוג A (סה"כ 12 נקודות)

תתי משימות 1,2,3 נמצאות בסוג A, שבהן  $n = 1$ .

לכל תת משימה יש את המגבלות הנוספות הבאות:

- תת משימה 1 (4 נקודות):  $m = 0$ .
- תת משימה 2 (4 נקודות):  $0 \leq m \leq 1$ .
- תת משימה 3 (4 נקודות):  $0 \leq m \leq 1000$ .

תת משימה מסוג B (סה"כ 54 נקודות)

תתי משימות 4,5,6 הם בסוג B, שבהן:

- $0 \leq n \leq 30, \frac{n}{2} \leq m \leq n^2$ .
- אין 2 מכתבים עם אותו שולח ומקבל.
- שמות כל החברים מופיעים בקלט של בוב (כלומר כל חבר שולח או מקבל לפחות מכתב אחד).

לכל תת משימה יש את המגבלות הנוספות הבאות:

- תת משימה 4 (24 נקודות):  $n = 26$ , שמות כל החברים הם אות אנגלית קטנה יחידה, ובקלט של אליס הם מופיעים לפי הסדר מ-a עד z.
- תת משימה 5 (24 נקודות):  $n = 26$ .
- תת משימה 6 (6 נקודות): ללא מגבלות מיוחדות.

תת משימה מסוג C (סה"כ 34 נקודות)

תתי משימות 7,8,9 הן מסוג C, בהן  $0 \leq m \leq 1000, 0 \leq n \leq 700$ .

לכל תת משימה יש את המגבלות הנוספות הבאות:

- תת משימה 7 (18 נקודות):  $n = 676$ , שמות כל החברים הם 2 אותיות אנגליות קטנות, ובקלט של אליס, הן מופיעות בסדר לקסיקוגרפי (כלומר, aa, ab, ac, ..., az, ba, ..., bz, ca, ..., zz).
- תת משימה 8 (10 נקודות):  $n = 676$ .
- תת משימה 9 (6 נקודות): ללא מגבלות נוספות.

## גרידר לדוגמה

הגרידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1:  $n \ m$
- שורה  $i$ :  $names_i \ numbers_i : (0 \leq i \leq n - 1)2 + i$
- שורה  $i$ :  $senders_i \ recipients_i : (0 \leq i \leq m - 1)2 + n + i$

הגרידר לדוגמה מוציא פלט בפורמט הבא:

- אם התוכנה מסיימת בהצלחה, הגריידר לדוגמה יפלוט  $n$  שורות, שכל אחת מהן מכילה מספר שלם, המייצג את התוצאה הסופית שחושבה על ידי הפונקציות שמימשתם לכל חבר.
- אחרת, הגריידר לא ידפיס כלום לstdout, וידפיס את הודעות השגיאה לקובץ abc.log בתיקייה.
- בנוסף, הגריידר לדוגמה יפלוט את הערכים של  $l_A, l_B, l$  וזמן הריצה של כל פונקצייה לabc.log.

הגרידר לדוגמה לא יבדוק את מגבלת הזיכרון, והמגבלה שעבור אותם  $n / m, l_A / l_B$  חייבים להיות שווים.