

D. משחק ניחושים

שם הבעיה	Guessing Game
מגבלת זמן	4 שניות
מגבלת זכרון	1 ג'יגהבייט

בעיר העתיקה של לונד, יש רחוב עם N בתים בשורה, הממוספרים מ-0 עד $N - 1$. אָמה גרה באחד מהבתים הללו והחברים שלה אנה וברטיל רוצים לגלות באיזה אחד. במקום פשוט לגלות לחברים שלה איפה היא גרה, אָמה החליטה לשחק איתם משחק. לפני שהמשחק מתחיל, אנה וברטיל יודעים רק את מספר הבתים ברחוב. בשלב זה, אנה וברטיל יכולים לבחור מספר שלם חיובי K ולהסכים על אסטרטגיה. כל תקשורת לאחר מכן אסורה.

המשחק עצמו בנוי משני שלבים. בשלב הראשון, אָמה בוחרת סדר לביקור הבתים, כך שהבית שלה הוא האחרון לבקר בו. לאחר מכן היא מובילה את אנה לבתים אחד אחר השני בסדר זה, בלי לגלות לאנה את הסדר מראש. לכל בית מלבד הבית של אָמה, לאנה מותר לכתוב מספר שלם יחיד בין 1 ל- K על דלת הכניסה של הבית עם חתיכת גיר. לבית האחרון שהן מבקרות בו, שהוא הבית של אָמה, אָמה רושמת מספר שלם בין 1 ל- K על הדלת בעצמה.

בשלב השני של המשחק, ברטיל הולך לאורך הרחוב מבית 0 לבית $N - 1$ וקורא את כל המספרים שנרשמו על הדלתות על ידי אנה ואָמה. הוא רוצה כעת לנחש באיזה בית אָמה גרה. יש לו שני ניסיונות לנחש נכונה והוא ואנה מנצחים במשחק אם הוא מצליח. אחרת, אָמה מנצחת במשחק.

האם את יכולה לתכנן אסטרטגיה שתבטיח שאנה וברטיל ינצחו במשחק? האסטרטגיה שלך תנוקד לפי הערך של K (ככל שהוא קטן יותר, יותר טוב).

מימוש

זוהי בעיית ריצות מרובות, כלומר התוכנית שלך תורץ מספר פעמים. בפעם הראשונה שהיא תורץ, היא תממש את האסטרטגיה של אנה. אחר כך היא תממש את האסטרטגיה של ברטיל.

שורת הקלט הראשונה תכיל שני מספרים שלמים, P ו- N , כש- P הוא 1 או 2 (השלב הראשון או השני), ו- N הוא מספר הבתים. מלבד הקלט לדוגמה (שלא משתמשים בו לניקוד), N תמיד יהיה שווה 100 000.

הקלט לאחר מכן תלוי בשלב:

שלב 1

על התוכנית שלך להתחיל בהדפסת המספר K בשורה יחידה ($1 \leq K \leq 1\,000\,000$). לאחר מכן, $N - 1$ פעמים, עליה לקרוא שורה המכילה אינדקס i ($0 \leq i < N$), ולהדפיס שורה עם מספר שלם A_i ($1 \leq A_i \leq K$), כאשר A_i הוא

המספר שאנה רושמת על הדלת של הבית i . כל אינדקס i חוץ מהאינדקס של הבית של אָמה יופיע בדיוק פעם אחת, בסדר כלשהו שנקבע על ידי הגריידר.

שלב 2

על התוכנית שלך לקרוא שורה עם N מספרים שלמים, A_0, A_1, \dots, A_{N-1} , כש- A_i הוא המספר שרשום על הדלת של בית i .

לאחר מכן, עליה להדפיס שורה עם שני מספרים שלמים, s_1 ו- s_2 ($0 \leq s_i < N$), האינדקסים שמנחשים. מותר ש- s_1 ו- s_2 יהיו שווים.

פרטי מימוש

שימי לב שבעת הרצת התוכנית שלך בשלב 2, התוכנית מתחילה מחדש. זה אומר שאת לא יכולה לשמור את המידע במשתנים כלשהם בין ההרצות.

אחרי כל שורה שאת מדפיסה, וודאי שאת עושה flush ל-output standard, אחרת התוכנית שלך עלולה לקבל את המשוב Exceeded Limit Time. בפיתון, `print()` עושה flush באופן אוטומטי. ב-C++, הפקודה `cout << endl;` גם כן עושה flush בנוסף להדפסת שורה חדשה; אם את משתמשת ב-`printf`, השתמשי ב-`fflush(stdout)`.

הגריידר בבעיה זו יכול להיות אדפטיבי, כלומר הוא יכול לשנות את התנהגותו באופן שתלוי בפלט של התוכנית שלך, כדי למנוע מפתרונות יוריסטיים להתקבל. הוא יכול להריץ כריצת נסיון את שלב 1, להסתכל על הפלט שלך, ואז להריץ את שלב 1 באמת תוך שימוש במידע שהוא חילץ מההרצה הקודמת.

התוכנית שלך חייבת להיות דטרמיניסטית, כלומר להתנהג אותו הדבר אם היא מורצת פעמיים על אותו הקלט. אם את רוצה להשתמש באקראיות בתוכנית שלך, וודאי שאת משתמשת ב-`seed` קבוע. ניתן לעשות זאת על ידי שליחת קבוע ל-`srand` (ב-C++) או ל-`random.seed` (בפיתון), או, אם משתמשים במחוללי המספרים האקראיים ב-C++11, ע"י ציון ה-`seed` כשמייצרים את מחולל המספרים האקראיים. בפרט, את לא יכולה להשתמש ב-`srand(time(NULL))` ב-C++. אם הגריידר מזהה שהתוכנית שלך לא דטרמיניסטית, היא תקבל את המשוב Wrong Answer.

אם *סכום* זמני הריצה של (עד 3) הריצות הנפרדות של התוכנית שלך חורג ממגבלת הזמן, ההגשה שלך תישפט כ-Time Limit Exceeded.

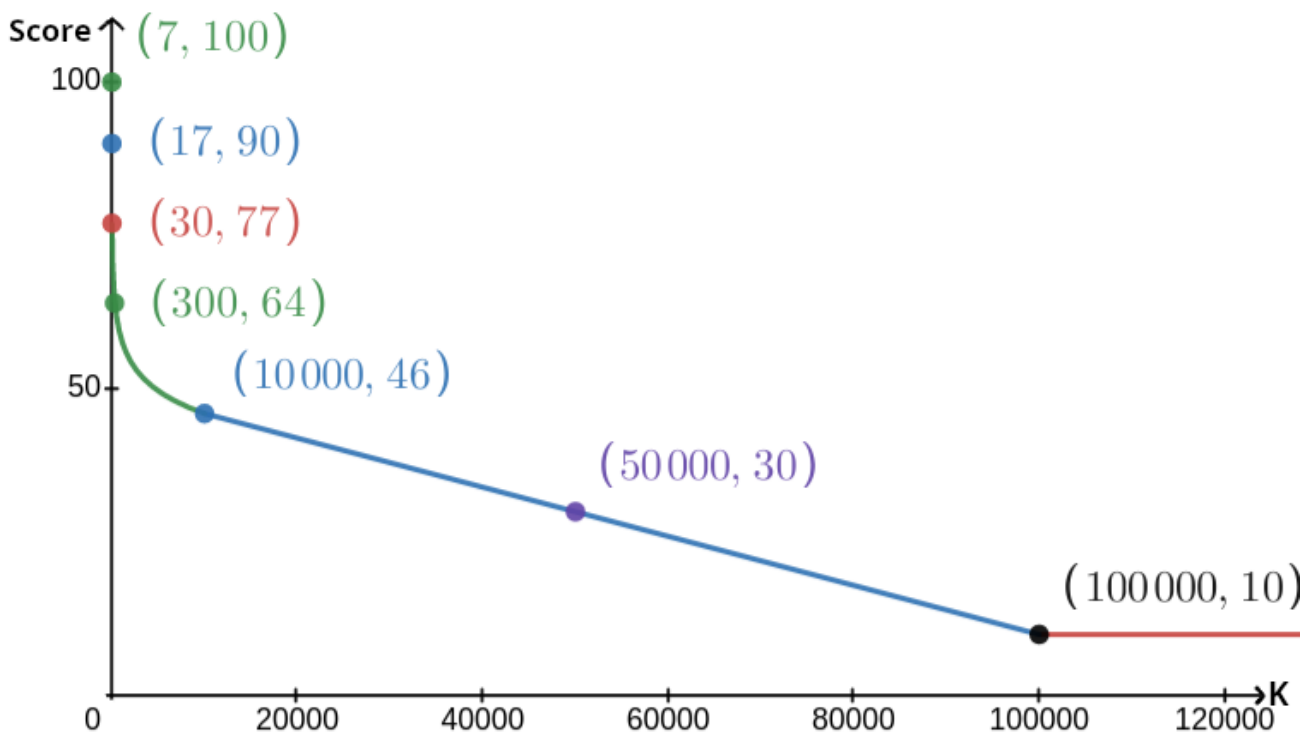
ניקוד

הפתרון שלך יבדק על מספר טסטקייסים. אם הפתרון שלך נכשל בלפחות אחד מהטסטקייסים האלו (למשל ע"י נתינת תשובות שגויות (Wrong Answer), קריסה (Run-Time Error), חריגה ממגבלת הזמן (Time Limit Exceeded), וכו'), את תקבלי 0 נקודות ואת המשוב המתאים.

אם הפתרון שלך מוצא את האינדקס של הבית של אָמה בכל הטסטקייסים, תקבלי את המשוב Accepted, וניקוד שיחושב כדלהלן. יהי K_{max} הערך המירבי של K השתמשת בו באחד הטסטקייסים. כתלות ב- K_{max} :

ניקוד	
10 נקודות	$K_{max} > 99\,998$
$10 + \lfloor 40(1 - K_{max}/10^5) \rfloor$ נקודות	$10\,000 < K_{max} \leq 99\,998$
$46 + \lfloor 31(4 - \log_{10}(K_{max})) / (4 - \log_{10}(30)) \rfloor$ נקודות	$30 < K_{max} \leq 10\,000$
$107 - K_{max}$ נקודות	$7 < K_{max} \leq 30$
100 נקודות	$K_{max} \leq 7$

פונקציית הניקוד מתוארת באיור מטה.



מהסטטיסטיס לדוגמה מתעלמים לצורך חישוב הניקוד, והפתרון שלך לא חייב לעבוד עבורו.

כלי בדיקה

כדי לסייע בבדיקת הפתרון שלך, אנו מספקות כלי פשוט שאת יכולה להוריד. ראי "attachments" בתחתית עמוד הבעיה ב-Kattis. הכלי הוא אופציונלי לשימוש, ומותר לך לשנות אותו. שימי לב שתוכנית הבדיקה הרשמית ב-Kattis שונה מכלי הבדיקה.

דוגמת שימוש (עם $N = 4$, $s = 2$, כש- s הוא המספר שנכתב על הבית האחרון שביקרו בו):

עבור תוכניות פייתון, נניח solution.py (בדרך כלל רצה כ-pypy3 solution.py):

```
python3 testing_tool.py pypy3 solution.py <<<"4 2"
```

עבור תוכניות C++, תחילה תקמפלי אותה (למשל עם `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) ואז הריצי:

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

כלי הבדיקה יבקר בבתים בסדר אקראי. כדי להשתמש בסדר ספציפי, שני את כלי הבדיקה במקום שרשום בו "MODIFY" "HERE".

אינטרקציה לדוגמה

מהטסטקייס לדוגמה מתעלמים לצורך חישוב הניקוד, והפתרון שלך לא חייב לעבוד עבורו.

הניחי שמתקיים $N = 4$ ושקמה גרה בבית 1. תהי A רשימת המספרים שרשומים על הבתים. תחילה, $A = [0, 0, 0, 0]$, כשהמשמעות של 0 היא ששום מספר לא נרשם על הבית המתאים.

בהרצה הראשונה של הקוד שלך:

$N = 4$ נתון. הפתרון שלך מגיב עם $K = 3$.

A_2 מבוקש. הפתרון שלך מגיב עם 3. A הוא כעת $[0, 0, 3, 0]$.

A_0 מבוקש. הפתרון שלך מגיב עם 1. A הוא כעת $[1, 0, 3, 0]$.

A_3 מבוקש. הפתרון שלך מגיב עם 2. A הוא כעת $[1, 0, 3, 2]$.

לבסוף, הגריידר קובע $A_1 = 2$, כך ש- $A = [1, 2, 3, 2]$ בסוף. זה מסמן את סוף השלב הראשון.

בשלב 2 של הקוד שלך, הפתרון שלך מקבל את הרשימה $1\ 2\ 3\ 2$.

הוא מגיב עם 3 1.

מכיוון שאחד מהניחושים שלך הוא האינדקס הנכון של הבית (1), אנה וברטיל מנצחים במשחק.

הפלט שלך	הפלט של הגריידר
	1 4
3	
	2
3	
	0
1	
	3
2	

הפלט שלך	הפלט של הגריידר
	2 4
	1 2 3 2
1 3	