



Paint By Numbers

המשחק שחור ופתור (Paint By Numbers) הוא משחק מחשבה מפורסם. נתבונן בגרסה פשוטה חד מימדית של המשחק. במשחק זה, נתונה לשחקן שורה של n תאים. התאים ממוספרים מ-0 עד $n - 1$ משמאל לימין. על השחקן לצבוע כל תא בשחור או בלבן. נשתמש ב-'X' כדי לסמן תאים שחורים וב-'_' כדי לסמן תאים לבנים.

לשחקן נתונה סדרה $c = [c_0, \dots, c_{k-1}]$ של k מספרים שלמים חיוביים: הרמזים. עליו לצבוע את התאים כך שהתאים השחורים בשורה ייצרו בדיוק k בלוקים של תאים עוקבים. בנוסף, מספר התאים השחורים בבלוק ה- i משמאל (המספור מתחיל מ-0) צריך להיות שווה ל- c_i . לדוגמה, אם הרמזים הם $c = [3, 4]$, הפתרון חייב להכיל בדיוק שני בלוקים של תאים שחורים עוקבים: אחד באורך 3 ואז אחד נוסף באורך 4. לכן, אם $n = 10$ ו- $c = [3, 4]$, פתרון אחד שמתאים לרמזים הוא "X X X _ X X X _". שימו לב ש-"X X X X _ X X X _" לא מתאים לרמזים כי הבלוקים של התאים השחורים אינם בסדר הנכון. כמו כן, "_ X X X X X X X _" לא מתאים לרמזים כי קיים בלוק יחיד של תאים שחורים, ולא שני בלוקים נפרדים.

נתון לכם משחק שחור ופתור שכבר נפתר באופן חלקי. כלומר, אתם יודעים את n ו- c , ובנוסף אתם יודעים שחלק מהתאים חייבים להיות שחורים וחלק מהתאים חייבים להיות לבנים. משימתכם היא להסיק מידע נוסף על התאים.

באופן יותר ספציפי, פתרון תקין הוא פתרון שמתאים לרמזים, ובנוסף מסכים עם הצבעים של התאים הידועים. תוכניתכם צריכה למצוא תאים שצבועים בשחור בכל פתרון תקין, ותאים שצבועים בלבן בכל פתרון תקין.

אתם יכולים להניח שקיים פתרון תקין אחד לפחות לקלט הנתון.

פרטי מימוש

עליכם לממש את הפונקציה הבאה (שיטה):

• `string solve_puzzle(string s, int[] c)`

◦ s : מחרוזת באורך n . לכל i ($0 \leq i \leq n - 1$) התו ה- i הוא:

◦ 'X' אם תא i חייב להיות שחור,

◦ '_' אם תא i חייב להיות לבן,

◦ '.' אם אין מידע על תא i .

◦ c : מערך בגודל k המכיל רמזים, כפי שהוגדר לעיל.

- הפונקציה צריכה להחזיר מחרוזת באורך n . לכל i ($0 \leq i \leq n - 1$) תו i במחרוזת הפלט צריך להיות:
 - 'X' אם תא i הוא שחור בכל פתרון תקין.
 - '_' אם תא i הוא לבן בכל פתרון תקין.
 - '?' אחרת (כלומר, אם קיימים שני פתרונות תקינים, כך שתא i הוא שחור באחד מהם ולבן באחר).

בשפת C, החתימה של הפונקציה מעט שונה:

• `void solve_puzzle(int n, char* s, int k, int* c, char* result)`

- n : אורך המחרוזת s (מספר התאים).
- k : אורך המערך c (מספר הרמזים).
- שאר הפרמטרים כפי שהוגדרו לעיל.
- במקום להחזיר מחרוזת של n תווים, הפונקציה צריכה לכתוב את התשובה למחרוזת `.result`.

קודי ה-ASCII של התווים בבעיה זו הם:

- 'X': 88,
- '_': 95,
- '!': 46,
- '?': 63.

אנא השתמשו בקבצי ה-`template` לפרטי המימוש עבור שפת התכנות שלכם.

דוגמאות

דוגמה 1

`solve_puzzle(".....", [3, 4])`

להלן הפתרונות התקינים למשחק:

- "XXX_XXXX_"
- "XXX__XXXX_"
- "XXX___XXXX"
- "_XXX_XXXX_"
- "_XXX__XXXX"
- "__XXX_XXXX"

ניתן לראות שהתאים באינדקסים 2, 6 ו-7 (המספור מתחיל מ-0) הם שחורים בכל פתרון תקין. כל שאר התאים יכולים, אבל לא חייבים, להיות שחורים. לכן, התשובה הנכונה היא "??X??X?".

דוגמה 2

`solve_puzzle(".....", [3, 4])`

בדוגמה זו הפתרון נקבע באופן יחיד והתשובה הנכונה היא "X X X _ X X X X".

דוגמה 3

`solve_puzzle("..._..._", [3])`

בדוגמה זו ניתן להסיק שתא מספר 4 חייב גם הוא להיות לבן - אין דרך להכניס שלושה תאים שחורים עוקבים בין התאים הלבנים באינדקסים 3 ו-5. לכן, התשובה הנכונה היא "??_??_??".

דוגמה 4

`solve_puzzle(".X.....", [3])`

יש רק שני פתרונות תקינים שמתאימים לתיאור:

"X X X _ _ _ _ _" •

"_ X X X _ _ _ _ _" •

לכן התשובה הנכונה היא "?? X X ? _ _ _ _ _".

תת משימות

בכל תת המשימות מתקיים $1 \leq k \leq n$, וגם $1 \leq c_i \leq n$ לכל $0 \leq i \leq k - 1$.

1. (7 נקודות): $n \leq 20$ וגם $k = 1$ וגם s כולל רק תווים של '!' (אין מידע)

2. (3 נקודות): $n \leq 20$ וגם s כולל רק תווים של '!'

3. (22 נקודות): $n \leq 100$ וגם s כולל רק תווים של '!'

4. (27 נקודות): $n \leq 100$ וגם s כולל רק '!' ו-'_' (קיים מידע רק על תאים לבנים)

5. (21 נקודות): $n \leq 100$

6. (10 נקודות): $n \leq 5000$ וגם $k \leq 100$

7. (10 נקודות): $n \leq 200000$ וגם $k \leq 100$

גרייזר לדוגמה

הגרייזר קורא את הקלט בפורמט הבא:

• שורה 1: המחרוזת s

• שורה 2: המספר השלם k ואחריו k מספרים שלמים c_0, \dots, c_{k-1}