

Вказівки щодо розв'язання завдання № 20 відбірково-тренувальних зборів команди міста Києва

1. Квадрат Рубика.

Назвемо «симетричною четвіркою» будь-яку з четвірок чисел, що стоять у клітинках з координатами (x, y) , $(2n+1-x, y)$, $(x, 2n+1-y)$, $(2n+1-x, 2n+1-y)$ (перша координата є номером стовпчика клітинки, друга — номером рядка) для довільних натуральних x та y , що не перевищують $2n$. Необхідною умовою того, що квадрат можна звести до потрібного вигляду є однаковість всіх симетричних четвірок у початковому розташуванні чисел в квадраті Рубика. Наприклад, набір чисел, що стоять у лівій верхній, правій верхній, лівій нижній та правій нижній комірках, має збігатися з набором чисел, що стоять у клітинках з координатами $(2, 3)$, $(2n-1, 3)$, $(2, 2n-2)$, $(2n-1, 2n-2)$. Це впливає з того, що число, яке спочатку знаходилося в комірці з координатами (x, y) , після довільної кількості операцій може опинитися лише в одній з комірок $(2n+1-x, y)$, $(x, 2n+1-y)$, $(2n+1-x, 2n+1-y)$ та самій (x, y) (а отже, набір чисел у кожній з n^2 симетричних четвірок — інваріант). У розташування чисел, до якого необхідно звести квадрат, симетричні четвірки, вочевидь, однакові. Значить, вони мають бути однаковими і для будь-якого розташування чисел, яке можна звести до необхідного.

Покажемо тепер, яким чином можна циклічно переставити довільні три з чотирьох чисел симетричної четвірки, всі інші числа квадрата залишивши при цьому на своїх місцях. Нехай, наприклад, три числа, що треба переставити, стоять у клітинках з координатами (x, y) , $(2n+1-x, y)$ та $(x, 2n+1-y)$ і циклічно переставити їх треба таким чином, щоб число, яке стояло в комірці (x, y) перейшло у клітинку з координатами $(2n+1-x, y)$. Тоді, провівши послідовно операції $-x, +y, -x, +y$ ($-x$ — відображення стовпчика з номером x , $+y$ — відображення рядка з номером y), досягнемо бажаного результату. Незавжди зрозуміти, що обравши довільні x та y та одну з послідовностей операцій $-x, +y, -x, +y$ або $+y, -x, +y, -x$, ми зможемо довільним чином циклічно переставити будь-які три числа будь-якої симетричної четвірки, не зачіпаючи при цьому інших чисел квадрата.

Назвемо «впорядкованою симетричною четвіркою» симетричну четвірку, на якій зафіксовано порядок чисел. Скажімо, симетричною четвіркою (a, b, c, d) з координатами (x, y) , де x та y не перевищують n (тобто, комірка (x, y) лежить у лівій верхній чверті великого квадрата), будемо називати четвірку чисел:

- a , що записано в клітинці (x, y) ;
- b , що записано у комірці $(2n+1-x, y)$;
- c , що знаходиться в клітинці $(x, 2n+1-y)$;
- d , що записане в $(2n+1-x, 2n+1-y)$.

Із зазначеного вище впливає таке твердження. Необхідною умовою того, що квадрат можна звести до потрібного вигляду, є те, що набори чисел в кожній впорядкованій симетричній четвірці однакові. Будемо вважати, що всі ці набори чисел — $\{a, b, c, d\}$ причому числа a, b, c, d попарно різні (інші випадки

розглянемо згодом). Зафіксуємо впорядковану симетричну четвірку чисел з координатами $(1, 1)$ (див. вище). Нехай, для визначеності, це буде (a, b, c, d) . В силу того, що всі симетричні четвірки в певному розумінні симетричні і жодній з них не можна віддати якусь перевагу, можна вважати, що саме порядок симетричної четвірки $(1, 1)$ буде в кінцевому підсумку збережено. Отже, тепер наше завдання — привести всі впорядковані симетричні четвірки до вигляду (a, b, c, d) . Зважаючи на можливість циклічної перестановки трьох чисел четвірки, що не зачіпає інших чисел квадрата, в кожній впорядкованій симетричній четвірці незалежно від інших ми можемо виконувати таку операцію: обрати довільні три числа четвірки і циклічно їх переставити. Наприклад, з четвірки (a, b, c, d) можна отримати (c, b, d, a) або (d, b, a, c) (обрали перше, третє і четверте число). Всього існує $4!=24$ різних можливих впорядкованих четвірки з набору попарно різних чисел $\{a, b, c, d\}$.

Рівно половину, 12 із них, за допомогою циклічних переставлень трьох чисел можна звести до четвірки (a, b, c, d) . Крім тривіального варіанту (a, b, c, d) це: (a, c, d, b) , (a, d, b, c) , (b, a, d, c) , (b, c, a, d) , (b, d, c, a) , (c, a, b, d) , (c, b, d, a) , (c, d, a, b) , (d, a, c, b) , (d, b, a, c) , (d, c, b, a) . Інші ж 12 варіантів неможливо привести до такої четвірки, проте можливо звести до будь-якої з четвірок, яка отримана з четвірки (a, b, c, d) перестановкою двох чисел, наприклад, до четвірки (b, a, c, d) . Такими четвірками є (a, b, d, c) , (a, c, b, d) , (a, d, c, b) , (b, a, c, d) , (b, c, d, a) , (b, d, a, c) , (c, a, d, b) , (c, b, a, d) , (c, d, b, a) , (d, a, b, c) , (d, b, c, a) і (d, c, a, b) . Щоб звести такі четвірки до вигляду (a, b, c, d) , нам необхідно відобразити якийсь рядок або стовпчик таблиці й таким чином змінити розташування чисел у інших впорядкованих четвірках. Кожна впорядкована симетрична четвірка, отже, може перебувати у двох станах:

0, якщо, її можна звести до виду (a, b, c, d) циклічними перестановками трьох чисел;

1, якщо це неможливо.

При цьому відображення якогось стовпчика або рядка таблиці, що зачіпає четвірку, змінює її стан на протилежний.

Побудуємо ще одну таблицю, $n \times n$, що складається з нулів та одиниць. Число в комірці з координатами (x, y) — стан впорядкованої симетричної четвірки з координатами (x, y) для початкового розташування чисел у квадраті Рубика. У клітинці $(1, 1)$ щойно побудованої таблиці стоїть 0. Тепер задача зводиться до такої. За одну операцію можна всі числа якогось рядка або стовпчика нової таблиці змінити на протилежні (0 на 1, а 1 на 0). Чи можливо вказати таку послідовність операцій, після якої у всіх клітинках таблиці будуть стояти нулі?

Ця задача вирішується вже доволі просто. Спочатку зазначимо, що зміни чисел у рядках і стовпчиках можна виконувати у довільному порядку — від цього результат не зміниться. Крім того, жоден з рядків або стовпчиків немає сенсу міняти два рази або більше, адже зміна стовпчика або рядка два рази повертає таблицю до того самого розташування чисел, яке було б в ній, якби ми жодного разу цей рядок або стовпчик не міняли. Маємо право вважати, що кожен рядок і стовпчик було змінено рівно раз або не було змінено взагалі. Більш того, можемо

також вважати, що перший стовпчик не було змінено жодного разу (якби ми могли звести таблицю до нульової, змінивши перший стовпчик, то ми також могли б звести таблицю до нульової, не міняючи його — доведіть це самостійно). Отже, змінено було ті і тільки ті рядки, на перетині яких з першим стовпчиком стоять одиниці. Проведемо ці зміни. Тепер, якщо всі стовпчики складаються або цілком з нулів, або цілком з одиниць, ми, очевидно, зможемо привести таблицю до бажаного вигляду. Інакше ж це зробити неможливо, а значить, і звести квадрат до потрібного вигляду також неможливо.

Якщо всі симетричні четвірки складаються з набору $\{a, b, c, d\}$, принаймні два числа з якого рівні між собою (наприклад, $a = d$, і набір перетворюється у $\{a, b, c\}$), зведення квадрату до потрібного вигляду можливе завжди: з кожної впорядкованої симетричної четвірки вдасться отримати впорядковану четвірку $\{a, b, c, d\}$ за допомогою лише циклічних переставлень трійок чисел.

2. Гнізда

Описана в задачі структура підключень до Інтернету також відома як «Декартове дерево» (англійською *Cartesian tree*). Існує відомий алгоритм на основі динамічного програмування для побудови цього дерева за лінійний час. Опишемо його.

Будемо вважати гнізда відсортованими по горизонталі. Пронумеруємо гнізда зліва на право.

Нехай у нас є побудоване дерево для $K-1$ гнізда. Нехай r — корінь цього дерева. Нехай $\text{right}[x]$ — номер гнізда до якого протягнули вправа від гнізда x , $y[x]$ — висота (y -координата) гнізда x . Розглянемо таку послідовність:

$$a_1 = r, \quad a_2 = \text{right}[a_1], \quad a_3 = \text{right}[a_2], \quad \dots, \quad a_M = \text{right}[a_{M-1}].$$

Очевидно, що $y[a_M] < \dots < y[a_3] < y[a_2] < y[a_1]$.

Спробуємо розширити дерево додавши K -те гніздо з координатами $(x_0; y_0)$:

- якщо $y_0 < y[a_M]$, тоді очевидно, треба покласти $\text{right}[a_M] = K$. Таким чином ми отримаємо правильне декартове дерево для K перший гнізд;
- якщо $y[a_M] < y_0 < y[a_{M-1}]$, то для того, щоб підтримати структуру дерева, потрібно покласти $\text{right}[a_{M-1}] = K$, $\text{left}[K] = a_M$;
- якщо $y[a_{M-1}] < y_0 < y[a_{M-2}]$, тоді потрібно покласти $\text{right}[a_{M-2}] = K$, $\text{left}[K] = a_{M-1}$...

Решта випадків — аналогічні. Окремо виділимо випадок $y[a_1] < y_0$, коли нова вершина повинна стати коренем дерева. Тобто, для підтримання структури потрібно покласти $\text{left}[K] = a_1$.

Псевдокод поданого алгоритму такий.

Для $K := 2$ to N

// маємо побудоване дерево з перших $K-1$ вершин

// пробуємо розширити це дерево додавши вершину (x_0, y_0)

якщо $y[\text{root}] < y_0$

// Окремий випадок: K вища за теперішній корінь

$\text{left}[K] = \text{root}$

$\text{root} = K$ // тепер новий корінь

інакше

$v = K-1$; - сама права вершина попереднього дерева

// Піднімаємось вгору по дереву

while $y[v] < y_0$

$v = \text{parent}[v]$

// Тепер вершина v така, що: $y[v] > y_0 > y[\text{right}[v]]$

якщо для v визначена $\text{right}[v]$, тоді

$\text{left}[K] = \text{right}[v]$

$\text{right}[v] = K$

$\text{parent}[K] = v$

Оцінка складності

Поданий алгоритм працює за лінійний час. Це очевидно, враховуючи те, що $\text{parent}[]$ від кожної вершини береться не більше одного разу. Отже в сумі за всі ітерації цикл while справдиться не більше N разів.

Деталі реалізації

Поданий псевдокод простіше буде запрограмувати, якщо додати нульову вершину, що вища за всі інші. Тоді не прийдеться розглядати випадок, коли вершина K стає коренем дерева.