

Перегородка

Темур та його помічник Улугбек готують фокус для шоу *Узбекистан має талант*. Хитрощі полягають у тому, щоб Темур розв'язав таку **задачу розбиття**: розбити задану множину додатних цілих чисел на K непорожні групи таким чином, щоб сума елементів у кожній групі була однаковою. Іншими словами, кожне ціле число з множини має бути присвоєно рівно одній з K груп, а сума в кожній групі має бути рівною. Наприклад, якщо задана множина $[2, 1, 6, 4, 5]$ та $K = 3$, коректним розбиттям буде $[2, 4]$, $[1, 5]$ та $[6]$. У цьому випадку сума елементів у кожній групі дорівнює рівно 6.

Трюк виконується наступним чином:

- Улугбек і Темур заходять в окремі кабінки і не можуть спілкуватися один з одним.
- Член журі надає Улугбеку масив A з N додатних цілих чисел, $A[0], A[1], \dots, A[N - 1]$, де кожне значення знаходиться в діапазоні від 1 до M включно. Член журі також надає Улугбеку значення K .
- Улугбек вибирає щонайбільше $K - 1$ (не обов'язково різних) цілих чисел для **додавання як нових елементів** до масиву A . Кожне додане ціле число також має бути в межах від 1 до M включно.
- Журі додає цілі числа, вибрані Улугбеком, до початкового масиву. Цей розширений масив потім **сортуються у неспадяючому порядку** та передається Темуру разом зі значенням K .
- Потім Темур повинен вирішити задачу розбиття для цього розширеного, відсортованого масиву.

Ваше завдання — розробити та імплементувати стратегію для Темура та Улугбека. Можна довести, що за заданих обмежень для них існує стратегія, яка дозволяє їм успішно вирішити задачу розбиття, незалежно від масиву A наданого журі.

Деталі реалізації

Вам слід реалізувати дві процедури.

Процедура, яку вам слід імплементувати для **Улугбека**, така:

```
std::vector<int> add_numbers(std::vector<int> A, int K, int M)
```

- A : масив довжини N , що представляє початковий масив, наданий Улугбеку.

- K : цільова кількість груп; зверніть увагу, що Улугбек може додати до масиву до $K - 1$ цілих чисел.
- M : максимально допустиме значення для кожного початкового та щойно доданого цілого числа.
- Ця процедура викликається рівно один раз для кожного тестового випадку.

Процедура повинна повернути масив C що містить цілі числа, які Улугбек хоче додати до початкового масиву. Нехай S позначає довжину масиву C .

- S має бути не більше ніж $K - 1$.
- Кожен елемент C має бути в межах від 1 до M включно.

Процедура, яку вам слід імплементувати для **Темура**, така:

```
std::vector<int> find_partition(std::vector<int> B, int K)
```

- B : масив довжини $N + S$ що містить як початкові цілі числа з A так і цілі числа, які додав Улугбек. Елементи масиву B відсортовані у неспадяючому порядку.
- K : цільова кількість груп.
- Ця процедура викликається рівно один раз для кожного тестового випадку.

Процедура повинна повертати масив P , який представляє розбиття B на K непересічні групи.

- Довжина P має бути $N + S$.
- Для кожного j такого, що $0 \leq j < N + S$, $P[j]$ означає групу, до якої належить $B[j]$.
- Групи повинні бути пронумеровані від 0 до $K - 1$, тобто $0 \leq P[j] < K$ має виконуватися для кожного $0 \leq j < N + S$.
- Для кожного i між 0 та $K - 1$ включно має бути принаймні один елемент j ($0 \leq j < N + S$) такий, що $P[j] = i$.
- Сума елементів, призначених кожній з груп K має бути ідентичною.

Під час фактичного оцінювання програма, яка викликає вищезазначені процедури, виконується **рівно два рази**.

- Під час першого запуску програми:
 - `add_numbers` викликається рівно один раз.
 - Повернений масив обробляється системою оцінювання для обчислення масиву B .
- Під час другого запуску програми:
 - `find_partition` викликається рівно один раз.

Обмеження

- $3 \leq N \leq 100\,000$

- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ для усіх i , таких, що $0 \leq i < N$.

Підзадачі

Підзадача	Бали	Додаткові обмеження
1	5	$N = 3$
2	4	$M = 1$
3	7	$M \leq 2$
4	10	$N \leq 10$
5	7	$K = 2$
6	12	$K \leq 3$
7	17	$K \leq 10$
8	20	$K \leq 100$
9	18	Без додаткових обмежень.

Приклад

Розглянемо наступний виклик:

```
add_numbers([8, 2, 9, 6, 1, 5, 5], 3, 9)
```

У цьому прикладі $A = [8, 2, 9, 6, 1, 5, 5]$. Ми хочемо розбити числа на $K = 3$ груп. Улугбек може додавати числа від 1 до $M = 9$.

Процедура може повернути $[5, 4]$, що означає, що Улугбек вирішує додати $S = 2$ нових цілих числа: 5 та 4. Це валідно, оскільки обидва елементи знаходяться в діапазоні від 1 до $M = 9$.

Розширений масив потім сортується та передається до Темура за допомогою наступного виклику:

```
find_partition([1, 2, 4, 5, 5, 5, 6, 8, 9], 3)
```

Ми можемо розділити цілі числа з цього масиву на три групи: $[1, 5, 9]$, $[2, 5, 8]$ та $[4, 5, 6]$. Зверніть увагу, що сума кожної з цих груп дорівнює 15. Щоб повідомити про цей розподіл, процедура повинна повернути масив $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Приклад градера

Формат вхідних даних:

```
N K M
A[0] A[1] ... A[N-1]
```

Формат вихідних даних:

Після завершення виклику `add_numbers`, приклад градера:

- Обчислює відсортований масив B .
- Виводить масиви C та B у такому форматі, **за якими йде порожній рядок**:

```
S
C[0] C[1] ... C[S-1]
B[0] B[1] ... B[N+S-1]
```

Після завершення виклику `find_partition`, градер виводить:

```
L
P[0] P[1] ... P[L-1]
```

Тут L — це довжина масиву P повернутого процедурою `find_partition`.