

Разбиване

Темур и асистента му, Улуг'бек, подготвят магически фокус за шоуто *Узбекистан търси талант*. Фокусът е свързан с това, Темур да реши следната **задача за разбиване**: да се разбие дадена колекция от цели положителни числа на K непразни групи, такива, че сумите на елементите във всяка група да са равни. С други думи, всяко число от колекцията трябва да бъде поставено в точно една от K групи и сумата на всяка група да е една и съща. Например, ако дадената колекция е $[2, 1, 6, 4, 5]$ и $K = 3$, валидно разбиване би било $[2, 4]$, $[1, 5]$, и $[6]$. В това разбиване, сумата на елементите във всяка група е точно 6.

Фокусът ще се изпълни както следва:

- Улуг'бек и Темур влизат в отделни кабинки и не могат да си комуникират.
- Член на журито дава на Улуг'бек един голям масив A от N цели положителни числа, $A[0], A[1], \dots, A[N - 1]$, където всяко число е между 1 и M , включително. Този член му дава и стойността на K .
- Улуг'бек избира да добави най-много $K - 1$ (не задължително различни) числа **като нови елементи** към масива A . Всяко добавено число трябва също да е между 1 и M , включително.
- Журито добавя числата, избрани от Улуг'бек, към оригиналния масив. Този разширен масив след това бива **сортиран в намаляващ ред** и даден на Темур, заедно със стойността на K .
- Темур трябва да реши задачата за разбиването за разширения сортиран масив.

Вашата задача е да измислите и имплементирате стратегия за Темур и Улуг'бек. Може да бъде доказано, че при дадените ограничения съществува стратегия за тях, която им позволява успешно да решат задачата за разбиване, независимо от масива A , предоставен от журито.

Детайли по имплементацията

Трябва да имплементирате две функции.

Функцията, която трябва да имплементирате за **Улуг'бек**, е:

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

- A : масив с дължина N , представляващ оригиналния масив, даден на Улуг'бек.

- K : нужният брой групи; забележете, че Улуг'бек може да добави най-много $K - 1$ числа към масива.
- M : максималната разрешена стойност за всяко оригинално и новодобавено число.
- Тази функция ще бъде извикана точно веднъж на тест.

Функцията трябва да върне масив C , съдържащ числата, които Улуг'бек иска да добави към оригиналния масив. Нека S да бъде дължината на масива C .

- S трябва да бъде най-много $K - 1$.
- Всеки елемент на C трябва да бъде между 1 и M , включително.

Функцията, която трябва да имплементирате за **Темур**, е:

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

- B : масив с дължина $N + S$, съдържащ както оригиналните числа, така и тези, добавени от Улуг'бек. Елементите на масива B са сортирани в ненамаляващ ред.
- K : нужният брой групи.
- Тази функция ще бъде извикана точно веднъж на тест.

Функцията трябва да върне масив P , представящ разбиването на B на K отделни групи.

- Дължината на P трябва да бъде $N + S$.
- За всяко j , където $0 \leq j < N + S$, $P[j]$ означава групата, в която $B[j]$ е поставено.
- Групите трябва да бъдат номерирани от 0 до $K - 1$, т.е. $0 \leq P[j] < K$ трябва да важи за всяко $0 \leq j < N + S$.
- За всяко i от 0 и $K - 1$, включително, трябва да има поне един елемент j ($0 \leq j < N + S$), такъв, че $P[j] = i$.
- Сумата на елементите, поставени във всяка от тези K групи, трябва да е една и съща.

В реалното оценяване, програма, която вика горните функции, се изпълнява **точно два пъти**.

- При първото изпълнение на програмата:
 - `add_numbers` се извиква точно веднъж.
 - Върнатият масив се обработва от оценяващата система, за да се получи масивът B .
- При второто изпълнение на програмата:
 - `find_partition` се извиква точно веднъж.

Ограничения

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$

- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ за всяко i , където $0 \leq i < N$.

Подзадачи

Подзадача	Точки	Допълнителни ограничения
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	Няма допълнителни ограничения.

Пример

Нека разгледаме следното извикване:

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

В този пример $A = [8, 2, 9, 6, 1, 5, 5]$. Искаме да разбием числата на $K = 3$ групи. Улуг'бек може да добавя числа между 1 и $M = 9$.

Функцията може да върне $[5, 4]$, което означава, че Улуг'бек е решил да добави $S = 2$ нови числа: 5 и 4. Те са валидни, защото и двете са между 1 и $M = 9$.

Тогава, разширеният масив е сортиран, и се дава на Темур със следното извикване:

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

Тук можем да разбием числата в този масив на следните три групи: $[1, 5, 9]$, $[2, 5, 8]$, и $[4, 5, 6]$. Забележете, че сумата на всяка от групите е равна на 15. За да съобщи това разбиване, функцията трябва да върне масива $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Локален грейдър

Формат на входа:

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

Формат на изхода:

След като извикването на `add_numbers` завърши, локалният грейдър:

- Изчислява сортирания масив B .
- Извежда масиви C и B в следния формат, **последвани от празен ред**:

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

След като извикването на `find_partition` завърши, локалният грейдър извежда:

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

Тук L е дължината на масива P , върнат от `find_partition`.