

Delenie

Temur a jeho asistent Ulug'bek pripravujú kúzelnický trik pre reláciu *Uzbekistan má talent*. Trik spočíva v tom, že Temur vyrieši nasledujúcu **deliacu úlohu**: rozdeľ danú sadu kladných celých čísel do K neprázdnych skupín tak, aby súčet prvkov v každej skupine bol rovnaký. Inými slovami, každé celé číslo zo sady musí byť priradené do presne jednej z K skupín a súčet v rámci každej skupiny musí byť rovnaký. Napríklad, ak je daná sada čísel $[2, 1, 6, 4, 5]$ a $K = 3$, platným rozdelením by boli skupiny $[2, 4]$, $[1, 5]$ a $[6]$. V tomto prípade je súčet prvkov v každej skupine presne 6.

Trik sa vykonáva takto:

- Ulug'bek a Temur vstúpia do samostatných kabín a nemôžu medzi sebou komunikovať.
- Člen poroty poskytne Ulug'bekovi pole A obsahujúce N kladných celých čísel, $A[0], A[1], \dots, A[N - 1]$, pričom každá hodnota leží v rozmedzí od 1 do M vrátane. Člen poroty tiež dá Ulug'bekovi hodnotu K .
- Ulug'bek vyberie najviac $K - 1$ celých čísel (nemusia byť rôzne), ktoré chce **pridať ako nové prvky** do poľa A . Každé pridané celé číslo musí byť tiež v rozmedzí od 1 do M vrátane.
- Porota pridá čísla vybrané Ulug'bekom k pôvodnému poľu. Toto rozšírené pole sa potom **zoradí v neklesajúcom poradí** a odovzdá sa Temurovi spolu s hodnotou K .
- Temur musí následne vyriešiť problém rozdelenia pre toto rozšírené a zoradené pole.

Vašou úlohou je navrhnúť a implementovať stratégiu pre Temura a Ulug'beka. Dá sa dokázať, že pri daných podmienkach existuje stratégia, ktorá im umožní vytvoriť korektné delenie bez ohľadu na to, ako vyzerá pole A zadané porotou.

Implementačné detaily

Naprogramujte dve funkcie.

Funkcia, ktorú máte implementovať pre **Ulug'beka**, je:

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

- A : pole dĺžky N , ktoré reprezentuje pôvodné pole zadané Ulug'bekovi.
- K : cieľový počet skupín. Toto číslo tiež hovorí, že Ulug'bek môže do poľa pridať nanajvýš $K - 1$ celých čísel.
- M : maximálna povolená hodnota pre každé pôvodné aj novo pridané číslo.
- Táto funkcia sa zavolá raz pre každý testovací vstup.

Funkcia má vrátiť pole C obsahujúce kladné celé čísla, ktoré chce Ulug'bek pridať do pôvodného poľa. Nech S označuje dĺžku poľa C .

- S musí byť najviac $K - 1$.
- Každý prvok poľa C musí byť v rozmedzí od 1 do M vrátane.

Funkcia, ktorú máte implementovať pre **Temura**, je:

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

- B : pole dĺžky $N + S$, ktoré obsahuje aj pôvodné čísla z A , aj čísla, ktoré pridal Ulug'bek. Prvky poľa B sú zoradené v neklesajúcom poradí.
- K : cieľový počet skupín.
- Táto funkcia sa zavolá raz pre každý testovací vstup.

Funkcia má vrátiť pole P , ktoré reprezentuje rozdelenie B na K disjunktných skupín.

- Dĺžka poľa P má byť $N + S$.
- Pre každé j také, že $0 \leq j < N + S$, $P[j]$ označuje skupinu, do ktorej patrí $B[j]$.
- Skupiny sú očíslované od 0 po $K - 1$. Pre každé $0 \leq j < N + S$ platí, že $0 \leq P[j] < K$.
- Pre každé $0 \leq i < K$ musí existovať aspoň jeden prvok j ($0 \leq j < N + S$) taký, že $P[j] = i$.
- Suma prvkov priradených ku každej z K skupín musí byť rovnaká.

Pri samotnom hodnotení sa váš program, ktorý obsahuje uvedené funkcie, spustí **presne dvakrát**.

- Pri prvom spustení programu:
 - Funkcia `add_numbers` je zavolaná práve raz.
 - Vrátené pole je spracované hodnotiacim systémom, ktorý z neho vyrobí pole B .
- Pri druhom spustení programu:
 - Funkcia `find_partition` je zavolaná práve raz.

Obmedzenia

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ pre každé i také, že $0 \leq i < N$.

Podúlohy

Podúloha	Body	Dodatočné obmedzenia
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	Bez dodatočných obmedzení.

Ukázkový príklad

Pozrime sa na nasledujúce volanie:

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

V tomto príklade je $A = [8, 2, 9, 6, 1, 5, 5]$. Chceme čísla rozdeliť do $K = 3$ skupín. Ulug'bek môže pridať najviac 2 čísla v rozmedzí od 1 do $M = 9$.

Funkcia môže vrátiť pole $[5, 4]$, čo znamená, že Ulug'bek sa rozhodne pridať $S = 2$ nové celé čísla: 5 a 4. Tieto čísla sú platné, keďže obe ležia v intervale od 1 po $M = 9$.

Rozšírené pole sa následne zoradí a odovzdá sa Temurovi nasledujúcim volaním:

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

Celé čísla z tohto poľa môžeme rozdeliť do týchto troch skupín: $[1, 5, 9]$, $[2, 5, 8]$ a $[4, 5, 6]$. Všimnite si, že súčet každej z týchto skupín sa rovná 15. Výstupom funkcie bude pole $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Ukážkový testovač

Formát vstupu:

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

Formát výstupu:

Po dokončení volania funkcie `add_numbers` ukážkový testovač:

- Vypočíta zoradené pole B .
- Vypíše polia C a B v nasledujúcom formáte a za nimi vypíše **jeden prázdny riadok**:

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

Po dokončení volania funkcie `find_partition` ukážkový testovač vypíše na výstup:

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

L je dĺžka poľa P , ktoré vráti funkcia `find_partition`.