

Particija

Temur i njegov asistent, Ulug'bek, pripremaju mađioničarski trik za emisiju *Uzbekistan ima talenat*. Trik se vrti oko Temurovog rješavanja sljedećeg **problema particije**: particioniranje datog skupa pozitivnih cijelih brojeva na K neprazne grupe tako da je zbir elemenata u svakoj grupi isti. Drugim riječima, svaki cijeli broj iz kolekcije mora biti dodijeljen tačno jednoj od K grupa, a zbir unutar svake grupe mora biti jednak. Na primjer, ako je data kolekcija $[2, 1, 6, 4, 5]$ i $K = 3$, validna particija bi bila $[2, 4]$, $[1, 5]$ i $[6]$. U ovom slučaju, zbir elemenata u svakoj grupi je tačno 6.

Trik se izvodi na sljedeći način:

- Ulug'bek i Temur ulaze u odvojene kabine i ne mogu međusobno komunicirati.
- Član porote daje Ulug'beku niz od A od N pozitivnih cijelih brojeva, $A[0], A[1], \dots, A[N - 1]$, gdje je svaka vrijednost između 1 i M uključujući i njih. Član porote također daje Ulug'beku vrijednost od K .
- Ulug'bek bira najviše $K - 1$ (ne nužno različitih) cijelih brojeva koje će **dodati kao nove elemente** u niz A . Svaki dodani cijeli broj također mora biti između 1 i M , uključujući te brojeve.
- Porota dodaje cijele brojeve koje je odabrao Ulug'bek originalnom nizu. Ovaj prošireni niz se zatim **sortira u neopadajućem redoslijedu** i daje Temuru, zajedno s vrijednošću K .
- Temur tada mora riješiti problem particije za ovaj prošireni, sortirani niz.

Tvoj zadatak je da osmisliš i implementiraš strategiju za Temura i Ulug'beka. Može se dokazati da, uz zadate uslove, postoji strategija koja im omogućava da uspješno riješe problem particionisanja, bez obzira na niz A koji zada žiri.

Detalji implementacije

Trebali biste implementirati dvije procedure.

Postupak koji biste trebali primijeniti za **Ulug'beka** je sljedeći:

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

- A : niz dužine N koji predstavlja originalni niz dat Ulug'beku.
- K : ciljani broj grupa; imajte na umu da Ulug'bek može dodati do $K - 1$ cijelih brojeva u niz.
- M : maksimalna dozvoljena vrijednost za svaki originalni i novododani cijeli broj.
- Ova procedura se poziva tačno jednom za svaki testni slučaj.

Procedura treba da vrati niz C koji sadrži cijele brojeve koje Ulug'bek želi da doda originalnom nizu. Neka S označava dužinu niza C .

- S mora biti najviše $K - 1$.
- Svaki element C mora biti između 1 i M (uključujući i M).

Postupak koji biste trebali primijeniti za **Temur** je sljedeći:

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

- B : niz dužine $N + S$ koji sadrži i originalne cijele brojeve iz A i cijele brojeve koje je Ulug'bek dodao. Elementi niza B su sortirani u neopadajućem redoslijedu.
- K : ciljani broj grupa.
- Ova procedura se poziva tačno jednom za svaki testni slučaj.

Procedura treba da vrati niz P koji predstavlja particiju skupa B na K disjunktne (međusobno nepresijecajuće) grupa.

- Dužina P treba biti $N + S$.
- Za svaki j takav da je $0 \leq j < N + S$, $P[j]$ označava grupu kojoj $B[j]$ pripada.
- Grupe trebaju biti numerirane od 0 do $K - 1$, što znači da $0 \leq P[j] < K$ mora vrijediti za svaki $0 \leq j < N + S$.
- Za svaki i između 0 i $K - 1$ uključujući i njih, mora postojati barem jedan element j ($0 \leq j < N + S$) takav da je $P[j] = i$.
- Zbir elemenata dodijeljenih svakoj od K grupa mora biti identičan.

U samom ocjenjivanju, program koji poziva gore navedene procedure se izvršava **tačno dva puta**.

- Tokom prvog pokretanja programa:
 - `add_numbers` se poziva tačno jednom.
 - Vraćeni niz obrađuje sistem ocjenjivanja kako bi izračunao niz B .
- Tokom drugog izvršavanja programa:
 - `find_partition` se poziva tačno jednom.

Ograničenja

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ za svako i takvo da je $0 \leq i < N$.

Podzadaci

Podzadatak	Rezultat	Dodatna ograničenja
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	Nema dodatnih ograničenja.

Primjer

Razmotrite sljedeći poziv:

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

U ovom primjeru $A = [8, 2, 9, 6, 1, 5, 5]$. Želimo podijeliti brojeve u $K = 3$ grupe. Ulug'bek može sabirati brojeve između 1 i $M = 9$.

Procedura može vratiti $[5, 4]$, što znači da Ulug'bek odlučuje dodati $S = 2$ novih cijelih brojeva: 5 i 4. Ovo je validno jer su oba elementa između 1 i $M = 9$.

Prošireni niz se zatim sortira i prosljeđuje Temuru sljedećim pozivom:

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

Cijeli brojevi iz ovog niza mogli bismo podijeliti u ove tri grupe: $[1, 5, 9]$, $[2, 5, 8]$ i $[4, 5, 6]$. Imajte na umu da je zbir svake od ovih grupa jednak 15. Da bi se prijavila ova particija, procedura treba da vrati niz $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Primjer ocjenjivača

Format unosa:

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

Izlazni format:

Nakon što se poziv funkcije `add_numbers` završi, primjer ocjenjivača:

- Izračunava sortirani niz B .
- Ispisuje nizove C i B u sljedećem formatu, **nakon čega slijedi prazan red**:

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

Nakon što se poziv funkcije `find_partition` završi, ocjenjivač ispisuje:

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

Ovdje je L dužina niza P kojeg vraća funkcija `find_partition`.