

Partition

Temur i njegov asistent Ulug'bek pripremaju mađioničarski trik za emisiju Uzbekistan Got Talent. Trik se zasniva na tome da Temur riješi sljedeći **problem particioniranja**: podijeliti dati niz pozitivnih cijelih brojeva u K nepraznih grupa tako da je zbir elemenata u svakoj grupi isti. Drugim riječima, svaki cijeli broj iz niza mora biti dodijeljen tačno jednoj od K grupa, a zbir u svakoj grupi mora biti jednak. Na primjer, ako je dati niz $[2, 1, 6, 4, 5]$ i $K = 3$, ispravna particija bila bi $[2, 4]$, $[1, 5]$ 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 žirija daje Ulug'beku niz A od N pozitivnih cijelih brojeva, $A[0], A[1], \dots, A[N - 1]$, gdje je svaka vrijednost između 1 i M , uključujući krajnje vrijednosti. Član žirija također daje Ulug'beku vrijednost 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 krajnje vrijednosti.
- Žiri dodaje cijele brojeve koje je Ulug'bek odabrao na kraj originalnog niza. Ovaj prošireni niz se zatim sortira u neopadajućem redoslijedu i daje Temuru, zajedno s vrijednošću K .
- Temur zatim mora riješiti problem particioniranja za ovaj prošireni, sortirani niz.

Vaš zadatak je osmisliti i implementirati strategiju za Temura i Ulug'beka. Može se dokazati da, pod datim ograničenjima, postoji strategija koja im omogućava da uspješno riješe problem particioniranja bez obzira na niz A koji žiri zada.

Detalji implementacije

Trebate implementirati dvije funkcije.

Funkcija koju trebate implementirati za **Ulug'beka** je:

```
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 : ciljni broj grupa; imajte na umu da Ulug'bek može dodati najviše $K - 1$ cijelih brojeva u niz.
- M : najveća dozvoljena vrijednost za svaki originalni i novododani cijeli broj.
- Ova funkcija se poziva tačno jednom za svaki testni primjer.

Funkcija treba vratiti niz C koji sadrži cijele brojeve koje Ulug'bek želi dodati u originalni niz. Neka S označava dužinu niza C .

- S mora biti najviše $K - 1$.
- Svaki element niza C mora biti između 1 i M , uključujući krajnje vrijednosti.

Funkcija koju trebate implementirati za **Temura** je:

```
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 sortirani su u neopadajućem redoslijedu.
- K : ciljni broj grupa.
- Ova funkcija se poziva tačno jednom za svaki testni primjer.

Funkcija treba vratiti niz P koji predstavlja particiju niza B u K disjunktnih grupa.

- Dužina niza P treba biti $N + S$.
- Za svaki j takav da je $0 \leq j < N + S$, $P[j]$ označava grupu kojoj pripada $B[j]$.
- Grupe trebaju biti numerisane od 0 do $K - 1$, što znači da za svaki $0 \leq j < N + S$ mora važiti $0 \leq P[j] < K$.
- Za svaki i između 0 i $K - 1$, uključujući krajnje vrijednosti, 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 isti.

Tokom stvarnog ocjenjivanja, program koji poziva navedene funkcije pokreće se tačno dva puta.

- Tokom prvog pokretanja programa:
 - `add_numbers` se poziva tačno jednom.
 - Sistem za ocjenjivanje obrađuje vraćeni niz kako bi izračunao niz B .
- Tokom drugog pokretanja 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 svaki i takav da je $0 \leq i < N$.

Podzadaci

Podzadatak	Bodovi	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 je $A = [8, 2, 9, 6, 1, 5, 5]$. Želimo particionirati brojeve u $K = 3$ grupe. Ulug'bek može dodati brojeve između 1 i $M = 9$.

Funkcija može vratiti $[5, 4]$, što znači da Ulug'bek odlučuje dodati $S = 2$ nova cijela broja: 5 i 4. Oni su ispravni 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)
```

Cijele brojeve iz ovog niza možemo podijeliti u ove tri grupe: $[1, 5, 9]$, $[2, 5, 8]$ i $[4, 5, 6]$. Primijetite da je zbir svake od ovih grupa jednak 15. Da bi prijavila ovu particiju, funkcija treba vratiti niz $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Sample Grader

Format ulaza:

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

Format izlaza:

Nakon što se završi poziv funkcije `add_numbers`, sample grader:

- 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 završi poziv funkcije `find_partition`, grader ispisuje:

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

Ovdje je L dužina niza P koji vraća `find_partition`.