

Jaotus

Temur ja tema assistent Ulug'bek valmistavad saate *Uzbekistan Got Talent* jaoks ette mustkunstitrikki. Trikk seisneb selles, et Temur lahendab järgmise **jaotusülesande**: jaga antud positiivsete täisarvude kollektsioon K mittetühjaks rühmaks nii, et iga rühma elementide summa on sama. Teisisõnu, iga täisarv algsest kollektsioonist tuleb määrata täpselt ühte K rühmast ja iga rühma elementide summa peab olema võrdne iga teise rühma elementide summaga. Näiteks kui on antud kollektsioon $[2, 1, 6, 4, 5]$ ja $K = 3$, siis sobiv jaotus oleks $[2, 4]$, $[1, 5]$ ja $[6]$. Sel juhul on iga rühma elementide summa täpselt 6.

Trikk esitatakse järgmiselt:

- Ulug'bek ja Temur sisenevad eraldi kabiinidesse ja ei saa omavahel suhelda.
- Kohtunik annab Ulug'bekile massiivi A , mis koosneb N positiivsest täisarvust, $A[0], A[1], \dots, A[N-1]$, kus iga väärtus on lõigust 1 kuni M (kaasa arvatud). Kohtunik annab talle ka väärtuse K .
- Ulug'bek lisab massiivi A **uute elementidena** maksimaalselt $K - 1$ (mitte tingimata erinevat) täisarvu. Iga lisatav täisarv peab ka olema lõigust 1 kuni M (kaasa arvatud).
- Žürii lisab Ulug'beki valitud täisarvud algsele massiivile. Seejärel sorteeritakse see laiendatud massiiv **mittekahanevas järjekorras** ja antakse Temurile koos väärtusega K .
- Seejärel peab Temur lahendama selle laiendatud, sorteeritud massiivi jaotamise ülesande.

Sinu ülesanne on leida ja ellu viia strateegia, mida Temur ja Ulug'bek saaks kasutada. Saab tõestada, et antud piirangute raames on olemas strateegia, mis võimaldab neil jaotusülesande edukalt lahendada, sõltumata žürii poolt antud massiivist A .

Realisatsioon

Sul tuleb kirjutada kaks funktsiooni.

Ulug'beki jaoks tuleb kirjutada järgmine funktsioon:

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

- A : Massiiv pikkusega N , mis esindab Ulug'bekile antud algset kollektsiooni.
- K : Soovitud rühmade arv. Pane tähele, et Ulug'bek võib lisada massiivile kuni $K - 1$ arvu.
- M : Iga algse ja iga lisatava täisarvu maksimaalne lubatud väärtus.
- Seda funktsiooni kutsutakse igas testis välja täpselt üks kord.

Funktsioon peaks tagastama massiivi C , mis sisaldab täisarve, mida Ulug'bek soovib algele massiivile lisada. Olgu S massiivi C pikkus.

- S peab olema maksimaalselt $K - 1$.
- Iga C element peab olema lõigust 1 kuni M (kaasa arvatud).

Temuri jaoks tuleks kirjutada järgmine funktsioon:

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

- B : Massiiv pikkusega $N + S$, mis sisaldab nii algseid täisarve massiivist A kui ka täisarve, mille Ulug'bek lisas. Massiivi B elemendid on järjestatud mittekahanevas järjekorras.
- K : Soovitud rühmade arv.
- Seda funktsiooni kutsutakse igas testis välja täpselt üks kord.

Funktsioon peab tagastama massiivi P , mis kirjeldab B jaotust K eraldiseisvaks rühmaks.

- P pikkus peab olema $N + S$.
- Iga $0 \leq j < N + S$ korral näitab $P[j]$ rühma, kuhu $B[j]$ kuulub.
- Rühmad tuleb nummerdada 0 kuni $K - 1$, mis tähendab, et iga $0 \leq j < N + S$ puhul peab kehtima $0 \leq P[j] < K$.
- Iga i kohta lõigust 0 kuni $K - 1$ (kaasa arvatud) peab leiduma vähemalt üks element j ($0 \leq j < N + S$) nii, et $P[j] = i$.
- Kõigi K rühma elementide summad peavad olema võrdsed.

Tegeliku hindamise käigus käivitatakse programm, mis sisaldab ülaltoodud funktsioone, **täpselt kaks korda**.

- Programmi esimesel käivitamisel:
 - `add_numbers` kutsutakse välja täpselt üks kord.
 - Hindamissüsteem töötleb tagastatud massiivi, et arvutada massiiv B .
- Programmi teise käivitamise ajal:
 - `find_partition` kutsutakse välja täpselt üks kord.

Piirangud

- $3 \leq N \leq 100\,000$.
- $2 \leq K \leq 100\,000$.
- $K \leq N$.
- $1 \leq M \leq 10^9$.
- $1 \leq A[i] \leq M$ iga $0 \leq i < N$ korral.

Alamülesanded

Alamülesanne	Skoor	Lisapiirangud
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	Lisapiiranguid pole.

Näide

Vaatleme järgmist väljakutset:

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

Selles näites $A = [8, 2, 9, 6, 1, 5, 5]$. Me tahame jagada arvud $K = 3$ rühmaks. Ulug'bek võib kollektsiooni lisada arve lõigust 1 kuni $M = 9$.

Funktsioon võib tagastada $[5, 4]$, mis tähendab, et Ulug'bek otsustab lisada $S = 2$ uut arvu: 5 ja 4. Need sobivad, kuna mõlemad jäävad lõiku 1 kuni $M = 9$.

Seejärel laiendatud massiiv sorteeritakse ja edastatakse Temurile järgmise kutsega:

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

Me võime selle massiivi täisarvud jagada kolme rühma: $[1, 5, 9]$, $[2, 5, 8]$ ja $[4, 5, 6]$. Pane tähele, et iga rühma summa on 15. Selle jaotuse kuvamiseks peaks protseduur tagastama massiivi $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Näidishindaja

Sisendi vorming

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

Väljundi vorming

Pärast funktsiooni `add_numbers` kutset näidishindaja:

- Arvutab sorteeritud massiivi B .
- Prindib massiivid C ja B järgmises vormingus, **millele järgneb tühi rida**:

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

Pärast funktsiooni `find_partition` kutset annab hindaja välja järgmise tulemuse:

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

Siin L on funktsiooni `find_partition` tagastatud massiivi P pikkus.