

Sadalījums

Timurs un viņa asistents Ulugbeks gatavo burvju triku Uzbekistānas talantu šovam *Uzbekistan Got Talent*. Trikā Timurs risina **sadalīšanas uzdevumu**: sadalīt doto pozitīvu skaitļu kolekciju K netukšās grupās tā, lai katras grupas elementu summa būtu vienāda. Formāli, katrs vesels skaitlis no kolekcijas ir jāpiešķir tieši vienai no K grupām, un summai katrā grupā ir jābūt vienādai. Piemēram, ja dotā kolekcija ir $[2, 1, 6, 4, 5]$ un $K = 3$, derīgs sadalījums būtu $[2, 4]$, $[1, 5]$ un $[6]$. Šajā gadījumā katras grupas elementu summa ir tieši 6.

Triks tiek izpildīts šādi:

- Ulugbeks un Timurs ieiet atsevišķās kabīnēs un nevar sazināties savā starpā.
- Žūrijas loceklis iedod Ulugbekam masīvu A ar N naturāliem skaitļiem, $A[0], A[1], \dots, A[N-1]$, kur katra vērtība ir no 1 līdz M (ieskaitot). Žūrijas loceklis Ulugbekam iedod arī vērtību K .
- Ulugbeks izvēlas ne vairāk kā $K - 1$ (ne obligāti atšķirīgus) veselus skaitļus, ko **pievienot kā jaunus elementus** masīvam A . Katram pievienotajam veselajam skaitlim jābūt no 1 līdz M (ieskaitot).
- Žūrija pievieno Ulugbeka izvēlētos veselos skaitļus sākotnējam masīvam. Šis paplašinātais masīvs pēc tam tiek **sakārtots nedilstošā secībā** un dots Timuram kopā ar vērtību K .
- Pēc tam Timuram ir jāatrisina šī paplašinātā, sakārtotā masīva sadalīšanas uzdevums.

Tavs uzdevums ir izstrādāt un implementēt stratēģiju Timuram un Ulugbekam. Var pierādīt, ka, ievērojot dotos ierobežojumus, pastāv stratēģija, kas ļauj veiksmīgi atrisināt sadalīšanas uzdevumu neatkarīgi no žūrijas sniegtā masīva A .

Implementēšanas detaļas

Tev jāimplementē divas procedūras.

Procedūra, kas Tev ir jāimplementē **Ulugbekam**, ir:

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

- A : masīvs garumā N , kas apraksta Ulugbekam iedoto sākotnējo masīvu.
- K : nepieciešamais grupu skaits; ņemiet vērā, ka Ulugbeks var masīvam pievienot līdz $K - 1$ veselu skaitli.
- M : katra sākotnējā un no jauna pievienotā veselā skaitļa maksimālā atļautā vērtība.
- Šī procedūra katram testam tiek izsaukta tieši vienu reizi.

Šai procedūrai jāatgriež veselo skaitļu masīvu C , ko Ulugbeks grib pievienot sākotnējam masīvam. Ar S apzīmēsim masīva C garumu.

- S ir jābūt ne lielākam par $K - 1$.
- Katram C elementam ir jābūt starp 1 un M (ieskaitot).

Procedūra, kas Tev ir jāimplementē **Timuram**, ir:

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

- B : masīvs garumā $N + S$, kas satur gan veselos skaitļus no sākotnējā masīva A , gan veselos skaitļus, ko pievienoja Ulugbeks. Masīva B elementi ir sakārtoti nedilstošā secībā.
- K : nepieciešamais grupu skaits.
- Šī procedūra katram testam tiek izsaukta tieši vienu reizi.

Šai procedūrai jāatgriež masīvs P , kas apraksta masīva B sadalīšanu K nepārklājošās grupās.

- P garumam ir jābūt $N + S$.
- Katram j , kur $0 \leq j < N + S$, $P[j]$ apzīmē grupu, kurai pieder $B[j]$.
- Grupām ir jābūt numurētām no 0 līdz $K - 1$, jeb $0 \leq P[j] < K$ ir jāizpildās visiem $0 \leq j < N + S$.
- Katram i no 0 līdz $K - 1$ (ieskaitot), ir jābūt vismaz vienam elementam j ($0 \leq j < N + S$), tādām, ka $P[j] = i$.
- Katrai no K grupām piešķirto elementu summai ir jābūt vienādei.

Oficiālā vērtēšanā programma, kas izsauc iepriekš minētās procedūras, tiek izpildīta **tieši divas reizes**.

- Programmas pirmās izpildīšanas laikā:
 - `add_numbers` tiek izsaukta tieši vienu reizi.
 - Vērtētājs apstrādā atgrieztos masīvu, lai aprēķinātu masīvu B .
- Programmas otrās izpildīšanas laikā:
 - `find_partition` tiek izsaukta tieši vienu reizi.

Ierobežojumi

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

Apakšuzdevumi

Apakšuzdevums	Punkti	Papildu ierobežojumi
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 papildu ierobežojumiem.

Piemērs

Aplūkosim šādu procedūras izsaukumu:

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

Šajā piemērā $A = [8, 2, 9, 6, 1, 5, 5]$. Mēs gribam sadalīt skaitļus $K = 3$ grupās. Ulugbeks var pievienot skaitļus no 1 līdz $M = 9$.

Procedūra var atgriezt $[5, 4]$, kas nozīmē, ka Ulugbeks izlemj pievienot $S = 2$ jaunus veselus skaitļus: 5 un 4. Tas ir atļauts, jo abi elementi ir starp 1 un $M = 9$.

Tad paplašinātais masīvs tiek sakārtots un iedots Timuram ar šādu procedūras izsaukumu:

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

Mēs varētu sadalīt veselus skaitļus no šī masīva trijās grupās: $[1, 5, 9]$, $[2, 5, 8]$, un $[4, 5, 6]$. Ņemiet vērā, ka katras grupas elementu summa ir vienāda ar 15. Lai paziņotu šo sadalījumu, procedūrai ir jāatgriež masīvs $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Paraugvērtētājs

Ievaddatu formāts:

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

Izvaddatu formāts:

Pēc procedūras `add_numbers` izsaukuma pabeigšanas paraugvērtētājs:

- Izrēķina sakārtoto masīvu B .
- Izvada masīvus C un B šādā formātā, **beigās izvadot tukšu rindu**:

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

Pēc procedūras `find_partition` izsaukuma pabeigšanas paraugvērtētājs atgriež:

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

Šeit L ir `find_partition` procedūras atgrieztā masīva P garums.