

Ndarja

Temur dhe asistenti i tij, Ulug'bek, po përgatisin një truk magjik për shfaqjen Uzbekistan Got Talent. Truku varet zgjidhja e Temur-it në *problemit të ndarjes* : të ndahet një koleksion i dhënë numrash të plotë pozitivë në K grupe joboshe, të tilla që shuma e elementeve në secilin grup të jetë e njëjtë. Me fjalë të tjera, çdo numër i plotë nga koleksioni duhet t'i caktohet saktësisht njërit prej K grupeve, dhe shuma brenda secilit grup duhet të jetë e barabartë. Për shembull, nëse koleksioni i dhënë është $[2, 1, 6, 4, 5]$ dhe $K = 3$, një ndarje e vlefshme do të ishte $[2, 4]$, $[1, 5]$ dhe $[6]$. Në këtë rast, shuma e elementeve në secilin grup është saktësisht 6.

Truku kryhet si më poshtë:

- Ulug'bek dhe Temur hyjnë në kabina të veçanta dhe nuk mund të komunikojnë me njëri-tjetrin.
- Një anëtar i jurisë i jep Ulug'bek një varg A me N numra të plotë, $A[0], A[1], \dots, A[N - 1]$, ku secila vlerë është midis 1 dhe M , duke përfshirë edhe kufijtë, së bashku me vlerën e K -së.
- Ulug'bek zgjedh të shumtën $K - 1$ numra të plotë (jo domosdoshmërisht të ndryshëm) për t'i shtuar si elemente të reja në vargun A . Secili numër i plotë i shtuar duhet të jetë gjithashtu midis 1 dhe M , duke përfshirë edhe kufijtë.
- Juria ia shton numrat e plotë të zgjedhur nga Ulug'bek vargut origjinal. Ky varg i zgjeruar më pas renditet në rend jozbritës dhe i jepet Temur, së bashku me vlerën e K -së.
- Temur më pas duhet ta zgjidhë problemin e ndarjes për këtë varg të zgjeruar dhe të renditur.

Detyra juaj është të hartoni dhe të implementoni një strategji për Temur dhe Ulug'bek.

Detajet e implementimit

Ju duhet të implementoni dy procedura.

Procedura që duhet të implementoni për *Ulug'bek* është:

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

- A : një varg me gjatësi N që paraqet vargun origjinal që i është dhënë Temur.
- K : numri i synuar i grupeve.
- M : vlera maksimale e lejuar për secilin numër të plotë origjinal dhe të shtuar rishtazi.
- Kjo procedurë thirret saktësisht një herë për çdo rast testimi.

Procedura duhet të kthejë një varg C që përmban numrat e plotë që Ulug'bek dëshiron t'i shtojë në vargun origjinal. Le të shënojmë me S gjatësinë e vargut C .

- S duhet të jetë të shumtën $K - 1$.
- Secili element i C -së duhet të jetë midis 1 dhe M , duke përfshirë edhe kufijtë.

Procedura që duhet të implementoni për *Temur* është:

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

- B : një varg me gjatësi $N + S$ që përmban si numrat e plotë origjinalë nga A , ashtu edhe numrat e plotë që shtoi Ulug'bek. Elementet e vargut B janë të renditura në rend jozbritës.
- K : numri i synuar i grupeve.
- Kjo procedurë thirret saktësisht një herë për çdo rast testimi.

Procedura duhet të kthejë një varg P që paraqet ndarjen e B -së në K grupe disjunkte.

- Gjatësia e P -së duhet të jetë $N + S$.
- Për çdo j të tillë që $0 \leq j < N + S$, $P[j]$ tregon grupin të cilit i përket $B[j]$.
- Grupet duhet të numërohen nga 0 deri në $K - 1$, që do të thotë se $0 \leq P[j] < K$ duhet të vlejë për çdo $0 \leq j < N + S$.
- Për çdo i midis 0 dhe $K - 1$, duke përfshirë edhe kufijtë, duhet të ekzistojë të paktën një element j ($0 \leq j < N + S$) i tillë që $P[j] = i$.
- Shuma e elementeve të caktuara në secilin nga K grupet duhet të jetë identike.

Në vlerësimin real, një program që i thërret procedurat e mësipërme ekzekutohet *saktësisht dy herë*.

- Gjatë ekzekutimit të parë të programit:
 - `add_numbers` thirret saktësisht një herë.
 - Vargu i kthyer përpunohet nga sistemi i vlerësimit për të llogaritur vargun B .
- Gjatë ekzekutimit të dytë të programit:
 - `find_partition` thirret saktësisht një herë.

Kufizimet

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ për çdo i të tillë që $0 \leq i < N$.

Nëndetyrat

Nëndetyra	Pikët	Kufizime shtesë
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	Nuk ka kufizime shtesë.

Shembull

Shqyrtoni thirrjen e mëposhtme:

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

Në këtë shembull $A = [8, 2, 9, 6, 1, 5, 5]$. Ne duam t'i ndajmë numrat në $K = 3$ grupe. Ulug'bek mund të shtojë numra midis 1 dhe $M = 9$.

Procedura mund të kthejë $[5, 4]$, që do të thotë se Ulug'bek vendos të shtojë $S = 2$ numra të plotë të rinj: 5 dhe 4. Këta janë të vlefshëm, meqenëse të dy elementet janë midis 1 dhe $M = 9$.

Vargu i zgjeruar më pas renditet dhe i kalohet Temur me thirrjen e mëposhtme:

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

Ne mund t'i ndajmë numrat e plotë nga ky varg në këto tre grupe: $[1, 5, 9]$, $[2, 5, 8]$ dhe $[4, 5, 6]$. Vini re se shuma e secilit prej këtyre grupeve është e barabartë me 15. Për ta raportuar këtë ndarje, procedura duhet të kthejë vargun $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Vlerësuesi shembull

Formati i hyrjes:

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

Formati i daljes:

Pasi përfundon thirrja e `add_numbers`, vlerësuesi shembull:

- Llogarit vargun e renditur B .
- Printon vargjet C dhe B në formatin e mëposhtëm, të ndjekur nga një rresht bosh:

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

Pasi përfundon thirrja e `find_partition`, vlerësuesi nxjerr:

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

Këtu, L është gjatësia e vargut P të kthyer nga `find_partition`.