Бөлүм

Темур жана анын жардамчысы Улугбек *Өзбекстан таланттарды издейт* шоусуна сыйкырдуу трюк даярдап жатышат. Бул ыкма Темурдун **бөлүү маселесин** чечүүсүнүн айланасында жүрөт: берилген оң бүтүн сандар жыйнагын ар бир топтогу элементтердин суммасы бирдей болгондой кылып K бош эмес топторго бөлүү. Башкача айтканда, коллекциядагы ар бир бүтүн сан K топторунун бирине болушу керек жана ар бир топтун ичиндеги сумма барабар болушу керек.

Мисалы, эгерде берилген жыйнак $[2, 1, 6, 4, 5]$ жана $K = 3$ болсо, жарактуу бөлүм $[2, 4]$, $[1, 5]$ жана $[6]$ болот. Бул учурда, ар бир топтогу элементтердин суммасы так 6 болот.

Трюк төмөнкүдөй аткарылат:

- Улугбек менен Темур өзүнчө кабинага кирип, бири-бири менен байланыша алышпайт.
- Калыстар тобунун мүчөсү Улугбекке N оң бүтүн сандарынан турган A массивин берет, $A[0], A[1], \ldots, A[N-1]$, мында ар бир маани 1 жана M ортосунда болот. Калыстар тобунун мүчөсү ошондой эле Улугбекке K маанисин берет.
- Улугбек A массивине **жаңы элементтер** кошуу үчүн эң көп дегенде $K - 1$ (сөзсүз түрдө айырмаланбаган) бүтүн сандарды тандайт жана кошот. Ар бир кошулган бүтүн сан да 1 жана M ортосунда болушу керек.
- Калыстар тобу Улугбек тандаган бүтүн сандарды баштапкы массивге кошот. Андан кийин бул кеңейтилген массив **кемибеген тартипте** иреттейт, Темурга K мааниси менен кошо берилет.
- Андан кийин Темур бул кеңейтилген, иреттелген массив үчүн бөлүү маселесин чечиши керек.

Сиздин милдетиңиз - Темур менен Улугбек үчүн стратегия иштеп чыгуу жана ишке ашыруу. Берилген чектөөлөрдүн астында, калыстар тобу тарабынан берилген A массивине карабастан, бөлүү маселесин ийгиликтүү чечүүгө мүмкүндүк берген стратегия бар экенин далилдөөгө болот.

Ишке ашыруунун чоо-жайы

Сиз эки жол-жобону ишке ашырышыңыз керек.

Улугбек үчүн сиз төмөнкү жол-жобону колдонушуңуз керек:

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

- A : Улугбекке берилген баштапкы массивди билдирген N узундуктагы массив.
- K : топтордун максаттуу саны; Улугбек массивге $K-1$ чейинки бүтүн сандарды кошо аларын эске алыңыз.
- M : ар бир баштапкы жана жаңы кошулган бүтүн сан үчүн уруксат берилген максималдуу маани.
- Бул процедура ар бир тест учуру үчүн бир гана жолу чакырылат.

Процедура Улугбек баштапкы массивге кошкусу келген бүтүн сандарды камтыган C массивин кайтарышы керек. C массивинин узундугун S деп атайлы.

- S эң көп дегенде $K-1$ болушу керек.
- C ар бир элементи 1 жана M ортосунда болушу керек, аларды кошо алганда.

Темур үчүн сиз төмөнкү жол-жобону колдонушуңуз керек:

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

- B : A дан алынган баштапкы бүтүн сандарды жана Улугбек кошкон бүтүн сандарды камтыган $N+S$ узундуктагы массив. B массивинин элементтери кемибеген тартипте иреттелет.
- K : топтордун максаттуу саны.
- Бул процедура ар бир тест учуру үчүн бир гана жолу чакырылат.

Процедура B массивин K бөлүнгөн топторго бөлүүнү билдирген P массивин кайтарышы керек.

- P узундугу $N+S$ болушу керек.
- $0 \leq j < N+S$ болгондой ар бир j үчүн, $P[j]$ таандык болгон $B[j]$ топту билдирет.
- Топтор 0 дөн $K-1$ге чейин номерлениши керек, башкача айтканда, ар бир $0 \leq j < N+S$ үчүн $0 \leq P[j] < K$ туура келиши керек.
- 0 жана $K-1$ ортосундагы ар бир i үчүн, $P[j] = i$ болушу үчүн жок дегенде бир j ($0 \leq j < N+S$) элементи болушу керек.
- K топторунун ар бирине дайындалган элементтердин суммасы бирдей болушу керек.

Чыныгы баалоодо, жогорудагы процедураларды чакырган программа **туура эки жолу** аткарылат.

- Программанын биринчи иштетилишинде:
  - `add_numbers` бир гана жолу чакырылат.
  - Кайтарылган массив B массивин эсептөө үчүн баалоо системасы тарабынан иштетилет.
- Программанын экинчи этабында:
  - `find_partition` ...

Чектөөлөр

- $3 \le N \le 100\,000$
- $2 \le K \le 100\,000$
- $K \le N$
- $1 \le M \le 10^9$
- $0 \le i < N$ болгондо ар бир i үчүн $1 \le A[i] \le M$.

Кошумча тапшырмалар

Кошумча тапшырма	Баалоо	Кошумча чектөөлөр
1	5	$N = 3$
2	4	$M = 1$
3	7	$M \le 2$
4	10	$N \le 10$
5	7	$K = 2$
6	12	$K \le 3$
7	17	$K \le 10$
8	20	$K \le 100$
9	18	Кошумча чектөөлөр жок.

Мисал

Төмөнкү чакырууну карап көрүңүз:

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

Бул мисалда $A = [8, 2, 9, 6, 1, 5, 5]$. Биз сандарды $K = 3$ топторуна бөлгүбүз келет. Улукбек 1 жана $M = 9$ ортосундагы сандарды кошо алат.

Процедура $[5, 4]$ кайтара алат, башкача айтканда, Улугбек $S = 2$ жаңы бүтүн сандарды кошууну чечет: 5 жана 4 . Булар жарактуу, анткени эки элемент тең 1 жана $M = 9$ ортосунда.

Андан кийин кеңейтилген массив сорттолуп, төмөнкү чакыруу менен Темурга өткөрүлүп берилет:

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

Бул массивдеги бүтүн сандарды үч топко бөлсөк болот: $[1, 5, 9]$, $[2, 5, 8]$ жана $[4, 5, 6]$. Бул топтордун ар биринин суммасы 15 барабар экенин эске алыңыз. Бул бөлүмдү билдирүү үчүн, процедура $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ массивин кайтарышы керек.

Үлгү грейдер

Киргизүү форматы:

```
NKM
A[0] A[1] ... A[N-1]
```

Чыгаруу форматы:

`add_numbers` чакыруу аяктагандан кийин, үлгү баалоочу:

- Сорттолгон B массивин эсептейт.
- C жана B массивдерин төмөнкү форматта басып чыгарат, **андан кийин бош сап**:

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

`find_partition` функциясын чакыруу аяктагандан кийин, баалоочу төмөнкүнү чыгарат:

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

Бул жерде, L - `find_partition` функциясы кайтарган P массивинин узундугу.