

Хуваалт

Төмөр ба түүний туслах Үлэгбэх нар *Авъяаслаг Узбекиуд* шоунд зориулан илбийн үзүүлбэр бэлдэж байгаа. Илбэ нь Төмөр дараах **хуваалтын бодлогыг** бодох тухай юм: өгөгдсөн бүхэл тоонуудын цуглуулгыг бүлэг бүрийн элементүүдийн нийлбэр нь ижил байхаар K ширхэг хоосон биш бүлэгт хуваах. Өөрөөр хэлбэл, цуглуулга дахь бүхэл тоо бүр K ширхэг бүлгийн яг нэгэнд нь харъяалагддаг ба бүлэг бүрийн элементүүдийн нийлбэр нь тэнцүү байна. Жишээ нь, хэрэв $[2, 1, 6, 4, 5]$ гэсэн цуглуулга өгөгдсөн ба $K = 3$ бол нэг зөв хуваалт нь $[2, 4]$, $[1, 5]$ ба $[6]$ байна. Энэ тохиолдолд бүлэг бүрийн элементүүдийн нийлбэр нь 6-тай тэнцүү байна.

Илбийг дараах байдлаар гүйцэтгэнэ:

- Үлэгбэх ба Төмөр өөр өөр өрөөнд орж хоорондоо харилцах боломжгүй болно.
- Шүүгч нарын нэг нь Үлэгбэхэд $A[0], A[1], \dots, A[N-1]$ гэсэн N ширхэг эерэг бүхэл тооноос тогтсон A массивыг өгнө. Мөн уг шүүгч Үлэгбэхэд K утгыг бас өгнө.
- Үлэгбэх A массив руу **шинэ элементүүд болгон нэмэх** дээд тал нь $K - 1$ ширхэг бүхэл тоог (ялгаатай байх албагүй) сонгоно. Нэмж байгаа бүхэл тоонуудын утга нь 1-ээс M (энэ тоог оруулаад) хүртлэх завсарт байна.
- Шүүгч Үлэгбэхийн сонгосон тоонуудыг анхны массив руу нэмнэ. Энэ өргөтгөсөн массивыг **үл буурахаар эрэмбэлээд** Төмөрт K утгын хамт өгнө.
- Дараа нь Төмөр энэ өргөтгөсөн, эрэмбэлэгдсэн массивын хувьд хуваалтын бодлогыг бодно.

Таны даалгавар бол Төмөр болон Үлэгбэх хоёрт зориулсан стратегийг санал болгон хэрэгжүүлэх явдал юм. Шүүгчийн өгсөн A массиваас үл хамааран, өгөгдсөн хязгаарлалтан дотор, тэд хуваалтын бодлогыг амжилттай шийдэхэд хүргэх стратеги оршин байгааг баталж болно.

Хэрэгжүүлэлтийн мэдээлэл

Та хоёр процедур хэрэгжүүлнэ.

Үлэгбэхэд зориулж хэрэгжүүлэх процедур доорх хэлбэртэй байна:

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

- A : Анх Үлэгбэхэд өгсөн массивыг илэрхийлэх N урттай массив.

- K : бүлгүүдийн тооны зорилтот утга, Үлэгбэх $K - 1$ хүртлэх тооны бүхэл тоог массивт нэмж болохыг анхаараарай.
- M : анх байсан ба шинээр нэмэгдсэн бүхэл тоонуудын зөвшөөрөгдсөн максимум утга.
- Уг процедурыг тест бүрийн хувьд яг нэг удаа дуудна.

Энэ процедур Үлэгбэхийн анхны массив руу нэмэхийг хүсч байгаа бүхэл тоонуудыг агуулсан C массивыг буцаана. S нь C массивын урт байг.

- S нь хамгийн ихдээ $K - 1$ гэсэн утгатай байна.
- C -ийн элемент бүрийн утга нь 1-ээс M (энэ тоог оруулаад) хүртлэх завсарт байна.

Төмөрт зориулж хэрэгжүүлэх процедур нь доорх хэлбэртэй байна:

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

- B : A массивт анх байсан бүхэл тоонууд ба Үлэгбэхийн нэмсэн бүхэл тоонуудыг агуулах $N + S$ урттай массив. B массивын элементүүд нь үл буурахаар эрэмбэлэгдсэн байна.
- K : бүлгүүдийн тооны зорилтот утга.
- Уг процедурыг тест бүрийн хувьд яг нэг удаа дуудна.

Уг процедур нь B -г K ширхэг үл огтлолцох бүлэгт хуваасныг илэрхийлэх P массивыг буцаана.

- P -гийн урт нь $N + S$ байна.
- $0 \leq j < N + S$ байх j бүрийн хувьд $P[j]$ нь $B[j]$ элемент аль группт харъяалагдахыг илэрхийлнэ.
- Бүлгүүдийг 0-ээс $K - 1$ хүртлэх тоонуудаар дугаарлах ба өөрөөр хэлбэл $0 \leq j < N + S$ утга бүрийн $0 \leq P[j] < K$ нөхцөл биелж байна.
- 0-ээс $K - 1$ (энэ тоог оруулаад) хүртлэх завсрын i бүрийн хувьд $P[j] = i$ нөхцөл биелж байх дор хаяж нэг j ($0 \leq j < N + S$) тоо оршин байна.
- K ширхэг групп тус бүрийн элементүүдийн нийлбэр нь ижилхэн байх ёстой.

Шалгагч ажиллах үедээ дээрх процедуруудыг дуудах программ **яг хоёр удаа** ажиллана.

- Программ эхний удаа ажиллах үед:
 - `add_numbers` процедурыг яг нэг удаа дуудна.
 - Буцаасан массивыг шалгагч боловсруулан B массивыг тооцоолж гаргана.
- Программ хоёр дахь удаа ажиллах үед:
 - `find_partition` процедурыг яг нэг удаа дуудна.

Хязгаарлалт

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$

- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ ($0 \leq i < N$ байх i бүрийн хувьд).

Дэд бодлого

Дэд бодлого	Оноо	Нэмэлт хязгаарлалт
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	Нэмэлт хязгаарлалт байхгүй.

Жишээ

Доорх дуудалтыг авч үзье:

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

Энэ жишээн дээр $A = [8, 2, 9, 6, 1, 5, 5]$ байна. Бид тоонуудыг $K = 3$ ширхэг бүлэгт хуваахыг хүсч байгаа. Үлэгбэх 1-ээс $M = 9$ хүртлэх завсраас тоонууд нэмж болно.

Уг процедур $[5, 4]$ тоонуудыг буцааж болох ба энэ нь Үлэгбэх $S = 2$ ширхэг бүхэл тоо болох 5 ба 4-ийг нэмэхээр шийдсэнийг илэрхийлнэ. Энэ хоёр тоо нь 1 ба $M = 9$ -ийн хооронд байгаа тул зөв үйлдэл юм.

Дараа нь өргөтгөсөн массивыг эрэмбэлээд доорх дуудалтыг ашиглан Төмөрт дамжуулна:

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

Бид энэ массив дахь тоонуудыг дараах гурван бүлэгт хувааж болно: $[1, 5, 9]$, $[2, 5, 8]$ ба $[4, 5, 6]$. Эдгээр бүлэг бүрийн элементүүдийн нийлбэр нь 15-тай тэнцүү байна. Энэ хуваалтыг илгээхийн тулд уг процедур $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ гэсэн массивыг буцаана.

Жишээ шалгагч

Оролтын формат:

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

Гаралтын формат:

`add_numbers` процедурын дуудалт ажиллаж дууссаны дараа жишээ шалгагч доорх зүйлсийг хийнэ:

- Эрэмбэлэгдсэн B массивыг тооцоолно.
- C ба B массивуудыг доорх форматаар, **хоосон мөрөөр тусгаарлан** хэвлэнэ:

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

`find_partition` процедурын дуудалт ажиллаж дууссаны дараа жишээ шалгагч дараахийг гаргана:

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

Энд, L нь `find_partition` процедурын буцаасан P массивын урт юм.