

التقسيم

يستعد عمر ومساعد هاشم لتقديم خدعة سحرية في برنامج *أوزبكستان غوت تالنت*. تدور الخدعة حول قيام عمر بحل مشكلة التقسيم التالية: تقسيم مجموعة معينة من الأعداد الصحيحة الموجبة إلى K مجموعة غير فارغة بحيث يكون مجموع العناصر في كل مجموعة متساويًا. بعبارة أخرى، يجب تخصيص كل عدد صحيح من المجموعة إلى مجموعة واحدة بالضبط من المجموعات K ، ويجب أن يكون المجموع داخل كل مجموعة متساويًا. على سبيل المثال، إذا كانت المجموعة المعطاة هي $[2, 1, 6, 4, 5]$ و $K = 3$ ، فإن التقسيم الصحيح سيكون $[2, 4]$ و $[1, 5]$ و $[6]$. في هذه الحالة، يكون مجموع العناصر في كل مجموعة هو بالضبط 6.

يتم تنفيذ الخدعة على النحو التالي:

- يدخل هاشم وعمر غرفتين منفصلتين ولا يمكنهما التواصل مع بعضهما البعض.
- يعطي أحد أعضاء لجنة التحكيم لهاشم مصفوفة A مكونة من N أعداد صحيحة موجبة، $A[0], A[1], \dots, A[N-1]$ ، حيث تقع كل قيمة بين 1 و M ضمناً. كما يعطي عضو لجنة التحكيم لهاشم القيمة K .
- يختار هاشم ما لا يزيد عن $K - 1$ (ليست بالضرورة أعداداً مختلفة) من الأعداد الصحيحة لإضافتها كعناصر جديدة إلى المصفوفة A . ويجب أيضاً أن يكون كل عدد صحيح مضاف بين 1 و M ضمناً.
- تقوم لجنة التحكيم بإلحاق الأعداد الصحيحة التي اختارها هاشم بالمصفوفة الأصلية. ثم يتم ترتيب هذه المصفوفة الموسعة بترتيب غير تنازلي وتسليمها إلى عمر، مع قيمة K .
- يجب على عمر بعد ذلك حل مشكلة التقسيم لهذه المصفوفة الموسعة والمرتبطة.

مهمتك هي ابتكار وتنفيذ استراتيجية لكل من عمر وهاشم. يمكن إثبات أنه، في ظل القيود المحددة، توجد استراتيجية تسمح لهما بحل مشكلة التقسيم بنجاح، بغض النظر عن المصفوفة A التي تقدمها لجنة التحكيم.

تفاصيل التنفيذ

يجب عليك تنفيذ إجراءين.

الإجراء الذي يجب عليك تنفيذه لـ هاشم هو:

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

- A : مصفوفة بطول N تمثل المصفوفة الأصلية المعطاة لهاشم.
- K : العدد المستهدف للمجموعات؛ لاحظ أن هاشم قد يضيف ما يصل إلى $K - 1$ عدداً صحيحاً إلى المصفوفة.
- M : الحد الأقصى المسموح به لكل عدد صحيح من الأعداد الأصلية والمضافة حديثاً.
- يتم استدعاء هذا الإجراء مرة واحدة فقط لكل حالة اختبار.

يجب أن يُرجع الإجراء مصفوفة C تحتوي على الأعداد الصحيحة التي يرغب هاشم في إضافتها إلى المصفوفة الأصلية. لنفترض أن S يمثل طول المصفوفة C .

- يجب أن يكون S لا يتجاوز $K - 1$.

- يجب أن يكون كل عنصر من عناصر C بين 1 و M ، بما في ذلك هذين العنصرين.

الإجراء الذي يجب عليك تنفيذه لـ **عمر** هو:

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

- B : مصفوفة بطول $N + S$ تحتوي على كل من الأعداد الصحيحة الأصلية من A والأعداد الصحيحة التي أضافها هاشم. يتم ترتيب عناصر المصفوفة B بترتيب غير تنازلي.
- K : العدد المستهدف من المجموعات.
- يتم استدعاء هذا الإجراء مرة واحدة فقط لكل حالة اختبار.

يجب أن يُرجع الإجراء مصفوفة P تمثل تقسيم B إلى K مجموعة منفصلة.

- يجب أن يكون طول P هو $N + S$.
- لكل j بحيث $0 \leq j < N + S$ ، فإن $P[j]$ يشير إلى المجموعة التي ينتمي إليها $B[j]$.
- يجب ترقيم المجموعات من 0 إلى $K - 1$ ، مما يعني أن $0 \leq P[j] < K$ يجب أن يتحقق لكل $0 \leq j < N + S$.
- لكل i بين 0 و $K - 1$ ضمناً، يجب أن يكون هناك عنصر واحد على الأقل j ($0 \leq j < N + S$) بحيث $P[j] = i$.
- يجب أن يكون مجموع العناصر المخصصة لكل مجموعة من المجموعات K متطابقاً.

في عملية التصحيح الفعلية، يتم تشغيل البرنامج الذي يستدعي الإجراءات المذكورة أعلاه مرتين بالضبط.

- أثناء التشغيل الأول للبرنامج:
 - يتم استدعاء `add_numbers` مرة واحدة بالضبط.
 - تتم معالجة المصفوفة المرجعة بواسطة نظام التصحيح لحساب المصفوفة B .
- أثناء التشغيل الثاني للبرنامج:
 - يتم استدعاء الدالة `find_partition` مرة واحدة بالضبط.

القيود

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

المهام الفرعية

المهمة الفرعية	النتيجة	القيود الإضافية
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	لا توجد قيود إضافية.

مثال

لنفترض الاستدعاء التالي:

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

في هذا المثال، $A = [8, 2, 9, 6, 1, 5, 5]$. نريد تقسيم الأعداد إلى $K = 3$ مجموعات. يمكن لهاشم إضافة أعداد تتراوح بين 1 و $M = 9$. يمكن أن يرجع الإجراء $[5, 4]$ ، مما يعني أن هاشم يقرر إضافة $S = 2$ أعداد صحيحة جديدة: 5 و 4. هذه الأعداد صالحة لأن كلا العنصرين يقعان بين 1 و $M = 9$.

ثم يتم ترتيب المصفوفة الموسعة وتمثيلها إلى عمر بالاستدعاء التالي:

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

يمكننا تقسيم الأعداد الصحيحة من هذه المصفوفة إلى هذه المجموعات الثلاث: $[1, 5, 9]$ ، $[2, 5, 8]$ ، و $[4, 5, 6]$. لاحظ أن مجموع كل مجموعة من هذه المجموعات يساوي 15. لإبلاغ هذا التقسيم، يجب أن تُرجع الإجراء المصفوفة $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

نموذج المصحح

تنسيق الإدخال:

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

تنسيق الإخراج:

بعد اكتمال استدعاء `add_numbers` ، يقوم مُصنّف العينة بما يلي:

- يحسب المصفوفة المُرَتبة B .
- يطبع المصفوفات C و B بالصيغة التالية، متبوعة بسطر فارغ:

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

بعد اكتمال استدعاء `find_partition` ، يُخرج برنامج التصحيح:

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

هنا، L هو طول المصفوفة P التي تُرجعها الدالة `find_partition` .