

Partisjonering

Temur og hans assistent, Ulug'bek, forbereder et trylletriks for *Usbekiske talenter*. Trikset innebærer at Temur løser følgende **partisjonsproblem**: partisjonere en gitt liste av positive heltall, ikke nødvendigvis ulike, inn i K ikke tomme grupper slik at summen av elementene i hver gruppe er identisk. Med andre ord, hvert heltall fra listen må bli tildelt nøyaktig én av de K gruppene, og summen i hver gruppe må være identisk. For eksempel, gitt listen $[2, 1, 6, 4, 5]$ og $K = 3$, ville $[2, 4]$, $[1, 5]$, og $[6]$ vært en gyldig partisjonering. Her summerer tallene i hver av gruppene til 6.

Triksset gjennomføres som følger:

- Ulug'bek og Temur går inn i hver sin boks hvor de ikke kan kommunisere med hverandre.
- Et jurymedlem gir Ulug'bek en liste A bestående av N positive heltall, $A[0], A[1], \dots, A[N-1]$, hvor hver verdi er mellom 1 og M inklusivt. Jurymedlemmet gir også Ulug'bek verdien til K .
- Ulug'bek velger høyst $K - 1$ (ikke nødvendigvis distinkte) heltall for å **legge til som nye elementer** til listen A . Hvert tall som legges til må være mellom 1 og M inklusivt.
- Juryen legger heltallene Ulug'bek valgte ut til den originale listen. Denne utvidede listen blir så **sortert i ikke-synkende rekkefølge** og gitt til Temur sammen med verdien av K .
- Temur må så løse partisjonsproblemet for den utvidede listen.

Oppgaven din er å lage og implementere en strategi for Temur og Ulug'bek. Det kan vises at det under de gitte betingelsene eksisterer en strategi som lar dem løse partisjonsproblemet uavhengig av listen A gitt av juryen.

Implementasjonsdetaljer

Du skal implementere to funksjoner.

Funksjonen du skal implementere for **Ulug'bek** er:

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

- A : en liste av lengde N som representerer den originale listen som gis til Ulug'bek.
- K : det ønskede antallet grupper. Merk at Ulug'bek kan legge maksimalt $K - 1$ heltall til listen.
- M : største verdien tillat for hvert originalt og nytt heltall.
- Funksjonen kalles nøyaktig en gang for hver test case.

Funksjonen skal returnere en liste C som inneholder heltallene Ulug'bek vil legge til den originale listen. La S være størrelsen til listen C

- S kan høyst være $K - 1$.
- Hvert element i C må være mellom 1 og M inklusivt.

Funksjonen du skal implementere for **Temur** er:

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

- B : en liste av lengde $N + S$ som inneholder både de opprinnelige heltallene fra A og heltallene som Ulug'bek la til. Elementene i listen B er sortert i ikke-synkende rekkefølge.
- K : det ønskede antallet grupper.
- Denne funksjonen kalles nøyaktig én gang for hvert testcase.

Funksjonen skal returnere en liste P som representerer partisjoneringen av B i K disjunkte grupper.

- Lengden på P skal være $N + S$.
- For hvert j slik at $0 \leq j < N + S$, angir $P[j]$ gruppen som $B[j]$ tilhører.
- Gruppene skal nummereres fra 0 til $K - 1$, som betyr at $0 \leq P[j] < K$ må gjelde for hver $0 \leq j < N + S$.
- For hver i mellom 0 og $K - 1$ inklusivt, må det finnes minst ett element j ($0 \leq j < N + S$) slik at $P[j] = i$.
- Summen av elementene som er tilordnet hver av de K gruppene må være identisk.

I selve gradingen kjøres et program som kaller funksjonene **nøyaktig to ganger**.

- Under den første kjøringen av programmet:
 - kalles «add_numbers» nøyaktig én gang.
 - Den returnerte listen behandles av graderen for å beregne listen « B ».
- Under den andre kjøringen av programmet:
 - kalles «find_partition» nøyaktig én gang.

Begrensninger

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

Deloppgaver

Deloppgave	Poeng	Ytterligere begrensninger
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	Ingen ytterligere begrensninger.

Eksempel

Gitt følgende kall:

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

Her er $A = [8, 2, 9, 6, 1, 5, 5]$. Vi ønsker å partisjonere heltallene inn i $K = 3$ grupper. Ulug'bek kan legge til heltall mellom 1 og $M = 9$

Funksjonen kan returnere « $[5, 4]$ », som betyr at Ulug'bek velger å legge til $S = 2$ nye heltall: 5 og 4. Disse er gyldige, siden begge elementene ligger mellom 1 og $M = 9$.

Den utvidede listen sorteres og blir sendt til Temur med følgende kall:

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

Vi kan dele heltallene i denne listen inn i følgende tre grupper: $[1, 5, 9]$, $[2, 5, 8]$ og $[4, 5, 6]$. Merk at summen av hver av disse gruppene er lik 15. For å rapportere denne inndelingen skal funksjonen returnere listen $[0, 1, 2, 0, 1, 2, 2, 1, 0]$.

Sample Grader

Inputformat:

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

Outputformat:

Etter `add_numbers`-kallet fullfører gjør sample graderen følgende:

- Lager den sorterte listen B .
- Printer listene C og B i følgende format, **etterfulgt av en tom linje**:

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

Etter `find_partition` kallet fullfører, skriver graderen ut:

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

Hvor L er lengden av listen P som returneres av `find_partition`.