



## Բացվածք

Կան  $N$  ոչ-բացասական ամբողջ թվեր՝  $a_1, a_2, \dots, a_N$ , որոնք բավարարում են հետևյալ անհավասարությանը.  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ : Ձիհակը ցանկանում է իմանալ *ամենամեծ հնարավոր* արժեքը  $a_{i+1} - a_i$  համար, որտեղ  $i$ -ն փոխվում է 1-ից մինչև  $N - 1$ : Մուտքային թվերը ուղղակիորեն չեն տրվելու Ձիհակի ծրագրին, այլ հասանելի են լինելու հատուկ ֆունկցիայի միջոցով: Մանրամասների համար տե՛ս քո նախընտրելի լեզվի համար իրականացման ենթաբաժինը:

## Խնդիր

Օգնեք Ձիհակին իրականացնելու ֆունկցիա, որը վերադարձնում է հնարավոր մեծագույն  $a_{i+1} - a_i$  արժեքը, որտեղ  $i$ -ն փոխվում է 1-ից մինչև  $N - 1$ :

## Իրականացումը C և C++ լեզուների համար

Պահանջվում է իրականացնել `findGap(T, N)` ֆունկցիան, որը վերցնում է հետևյալ պարամետրերը և վերադարձնում է `long long` տիպի ամբողջ թիվ.

- $T$  — ենթախնդրի համարը (1 կամ 2)
- $N$  — տրված ամբողջ թվերի քանակը:

Ձեր `findGap` ֆունկցիան կարող է կանչել `MinMax(s, t, &mn, &mx)` ֆունկցիան, որտեղ առաջին երկու `s` և `t` պարամետրերը `long long` տիպի ամբողջներ են, իսկ վերջին երկու `&mn` և `&mx` պարամետրերը `long long` տիպի հղումներ են, այսինքն, `mn`-ը և `mx`-ը `long long` տիպի փոփոխականներ են: Երբ `MinMax(s, t, &mn, &mx)`-ը կանչվում է, `mn` փոփոխականը ստանում է  $a_i$ -երից ամենափոքրի արժեքը, որը մեծ է կամ հավասար `s`-ից, իսկ `mx` փոփոխականը ստանում է ամենամեծ  $a_j$ -ն, որը փոքր է կամ հավասար `t`-ից: Եթե `s`-ի և `t`-ի միջև (ներառյալ իրենց) մուտքային թվեր չկան, ապա `mn`-ը և `mx`-ը, երկուսն էլ կստանան `-1` արժեքը: `MinMax`-ի կանչի ժամանակ `s`-ի արժեքը չպետք է մեծ լինի `t`-ի արժեքից: Եթե այս պայմանը չբավարարվի, ձեր ծրագիրը կավարտվի ոչ գրոյական ավարտի կոդով:

## Իրականացումը Պասկալ լեզվի համար

Պետք է իրականացնել `findGap(T, N)` ֆունկցիան, որը վերցնում է հետևյալ պարամետրերը և վերադարձնում է `Int64` տիպի ամբողջ թիվ.

- $T$  — ենթախնդրի համար (1 կամ 2) (`Integer` տիպի)
- $N$  — տրված ամբողջ թվերի քանակը (`LongInt` տիպի)

Ձեր `findGap` ֆունկցիան կարող է կանչել `MinMax(s, t, mn, mx)` պրոցեդուրան,

որտեղ առաջին երկու պարամետրերը՝  $s$ -ը և  $t$ -ն  $\text{Int64}$  տիպի փոփոխականներ են, իսկ վերջին երկու՝  $mn$  և  $mx$  պարամետրերը  $\text{Int64}$  տիպի **հղումով կանչվող** պարամետրեր են, այսինքն,  $mn$ -ը և  $mx$ -ը  $\text{Int64}$  տիպի փոփոխականներ են: Երբ  $\text{MinMax}(s, t, mn, mx)$  -ը ավարտվում է,  $mn$  փոփոխականը ստանում է ամենափոքր  $a_i$ -ի արժեքը, որը մեծ է կամ հավասար  $s$ -ից, իսկ  $mx$  փոփոխականը ստանում է ամենամեծ  $a_j$ -ն, որը  $t$ -ից փոքր է կամ հավասար: Եթե  $s$ -ի և  $t$ -ի միջև (ներառյալ) մուտքային թվեր չկան, ապա  $mn$ -ը և  $mx$ -ը՝ երկուսն էլ կստանան  $-1$  արժեքը:  $\text{MinMax}$ -ի կանչի ժամանակ  $s$ -ի արժեքը չպետք է մեծ լինի  $t$ -ի արժեքից: Եթե այդ պայմանը խախտվի, ձեր ծրագրի աշխատանքը կընդհատվի:

## Իրականացում բոլորի համար

Բացի ստանդարտ պահանջներից (ժամանակի և հիշողության սահմանափակումներ, կատարման ընթացքում սխալների բացառում և այլն), ձեր ծրագիրը պետք է բավարարի նաև հետևյալ պայմաններին.

- ձեր `findGap` ֆունկցիան պետք է վերադարձնի ճիշտ պատասխան,
- $\text{MinMax}$  ֆունկցիայի կանչերի հետ կապված  $M$  արժեքը չեպտք է գերազանցի թույլատրվող սահմանը (տես Միավորների հաշվարկ ենթաբաժինը):

## Օրինակ C, C++ համար

Դիտարկենք դեպք, որտեղ  $N = 4$  և  $a_1 = 2, a_2 = 3, a_3 = 6$ , և  $a_4 = 8$ :

Պատասխանը **3** է, և այն կարելի է հաշվել և վերադարձնել `findGap`-ի միջոցով կատարելով  $\text{MinMax}$ -ի հետևյալ կանչերը.

- կանչվում է  $\text{MinMax}(1, 2, \&mn, \&mx)$  -ը,  $mn$ -ը և  $mx$ -ը, երկուսն էլ ստանում են **2** արժեքը:
- կանչվում է  $\text{MinMax}(3, 7, \&mn, \&mx)$  -ը,  $mn$ -ը ստանում է **3**, իսկ  $mx$ -ը **6** արժեքը:
- կանչվում է  $\text{MinMax}(8, 9, \&mn, \&mx)$ , և  $mn$ -ն ու  $mx$ -ը, երկուսն էլ ստանում են **8** արժեքը:

## Օրինակ Պասկալ լեզվի համար

Դիտարկենք դեպք, որտեղ  $N = 4$  և  $a_1 = 2, a_2 = 3, a_3 = 6$ , և  $a_4 = 8$ :

Պատասխանը, որը **3** է, կարելի է ստանալ և վերադարձնել `findGap`-ի միջոցով, եթե կատարվեն  $\text{MinMax}$ -ի հետևյալ կանչերը.

- կանչվում է  $\text{MinMax}(1, 2, mn, mx)$  -ը, և  $mn$ -ը,  $mx$ -ը, երկուսն էլ ստանում են **2** արժեքը:
- կանչվում է  $\text{MinMax}(3, 7, mn, mx)$  -ը, և  $mn$ -ը ստանում է **3** արժեքը, իսկ  $mx$ -ը ստանում է **6** արժեքը:
- կանչվում է  $\text{MinMax}(8, 9, mn, mx)$  -ը, և  $mn$ -ը,  $mx$ -ը, երկուսն էլ ստանում են **8** արժեքը:

# Միավորների հաշվարկ

Բոլոր ենթախնդիրներում տեղի ունի  $2 \leq N \leq 100,000$  սահմանափակումը:

**Ենթախնդիր 1 (30 միավոր):** MinMax-ի յուրաքանչյուր կանչ  $M$ -ը կմեծացնի 1-ով: Ենթախնդրի համար լրվի միավորը կտրվի, եթե  $M \leq \frac{N+1}{2}$  բոլոր թեստերում:

**Ենթախնդիր 2 (70 միավոր):** Դիցուք  $k$ -ն MinMax-ի կանչի ժամանակ  $s$ -ից մեծ կամ հավասար և  $t$ -ից փոքր կամ հավասար մուտքային թվերի քանակն է: MinMax-ի յուրաքանչյուր կանչ  $M$ -ին կավելացնի  $k + 1$ : Վերջնական միավորը կհաշվարկվի համաձայն հետևյալ կանոնի. Ենթախնդրի վերջնական միավորը հավասար է բոլոր թեստերից ստացված միավորներից փոքրագույնին: Թեստի միավորը 70 է, եթե  $M \leq 3N$ , հակառակ դեպքում թեստի միավորը կլինի  $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ :

## Փարձարկումներ

Մտուգող համակարգից կարելի է ներբեռնել գրեյդրեի օրինակ, որը կարդում է տվյալները ստանդարտ մուտքից: Մուտքի առաջին տողը պետք է պարունակի երկու ամբողջ թիվ՝ ենթախնդրի  $T$  համարը և  $N$ -ը: Հաջորդ տողը պետք է պարունակի  $N$  ամբողջ թվեր դասավորված աճման կարգով: Գրեյդերի օրինակը ստանդարտ էլքում կարտածի findGap-ի վերադարձրած արժեքը և  $M$ -ի արժեքը, եթե մուտքային թեստը պատկանում է համապատասխան ենթախնդրին:

Հետևյալ մուտքային տվյալները նկարագրում են վերը բերված օրինակը.

```
2 4
2 3 6 8
```