

Počítanie jedlých húb (mushrooms)

Štefan je odborník na hriby, preto ho pozvali do Singapuru, aby im pomohol.

Štefan tam nazbieral n húb. Tie si očísloval od 0 po $n - 1$. Každá huba je jedného z dvoch typov: jedlá (typ A) alebo jedovatá (typ B).

Štefan za pomoci odvážneho domorodca experimentálne overil, že **huba s číslom 0 má typ A**. Momentálne nič netuší o typoch ostatných $n - 1$ húb, ktoré nazbieral. Navonok všetky nazbierané huby vyzerajú rovnako. Jediný spôsob, ktorým ich vie rozlíšiť, je použiť drahú mašinu v labáku.

Mašina na vstupe očakáva dve alebo viac húb uložených vedľa seba v jednom rade. Môžeme si ľubovoľne zvoliť, ktoré konkrétne huby chceme použiť a v ktorom poradí. Keď do mašiny uložíme nejaké huby, stačí stlačiť veľké zelené tlačidlo. Mašina sa zavrie, vykoná testy, a na ich základe nám oznámi práve jedno nesmierne užitočné číslo: počet **dvojíc susedných húb**, ktoré majú v rámci dvojice navzájom rôzny typ.

Ak by ste napríklad do tejto mašiny vložili štyri huby, ktoré majú postupne typy $[A, B, B, A]$ (v tomto poradí), mašina by na výstupe dala číslo 2.

Vašou úlohou je vymyslieť, ako pomocou Štefanovej mašiny zistiť, **koľko jedlých húb** (húb typu A) nazbieral.

Porovnávať huby v mašine je drahé a my nemáme neobmedzený rozpočet. Preto vo vašom riešení celkový počet húb umiestnených do mašiny (presnejšie, súčet počtov húb v mašine pri všetkých jej spusteniach) nesmie prekročiť 100 000.

Mašinu smiete spustiť nanajvýš 20 000-krát. Počet bodov vášho riešenia bude závisieť od toho, koľkokrát mašinu naozaj spustíte.

Detaily implementácie

Naprogramujte nasledujúcu funkciu:

```
int count_mushrooms(int n)
```

- n : počet húb.
- Táto funkcia má vrátiť počet húb typu A.
- Grader zavolá túto funkciu práve raz.

Vaša funkcia môže postupne viackrát volať nasledujúcu funkciu gradera:

```
int use_machine(int[] x)
```

- x : pole dĺžky od 2 po n , obsahujúce indexy húb v poradí, v akom ich chcete vložiť do mašiny.
- Prvky poľa x musia byť **navzájom rôzne** celé čísla od 0 po $n - 1$, vrátane.
- Túto funkciu smiete zavolať nanajvýš 20 000-krát.
- Keď sčítame dĺžky poľa x cez všetky volania funkcie `use_machine`, výsledok nesmie prekročiť 100 000.

Ukážkové príklady

Príklad 1

Predpokladajme, že Štefan nazbieral tri huby typov $[A, B, B]$, v tomto poradí. Grader by zavolať vašu funkciu nasledovne:

```
count_mushrooms(3)
```

Vaša funkcia teraz môže skúmať huby postupnými volaniami funkcie `use_machine`. Predpokladajme, že zavolať `use_machine([0, 1, 2])`. V tomto prípade by toto volanie vrátilo hodnotu 1.

Následne mohla vaša funkcia zavolať `use_machine([2, 1])`. Toto volanie by vrátilo 0.

V tejto chvíli by už vaša funkcia mala mať dosť informácií na to, aby vypočítala, že huba typu A je práve jedna (huba 0). Vaša funkcia `count_mushrooms` by teda mala vrátiť hodnotu 1.

Príklad 2

Majme tentokrát štyri huby, postupne typov $[A, B, A, A]$. Grader tento raz zavolať vašu funkciu nasledovne:

```
count_mushrooms(4)
```

Predpokladajme, že vaša funkcia postupne spravila dve volania: najskôr zavolať `use_machine([0, 2, 1, 3])` (dostala návratovú hodnotu 2) a potom zavolať `use_machine([1, 2])` (návratová hodnota 1).

V tejto chvíli zo získaných informácií síce ešte nevieme presne povedať, ktoré huby sú typu A, môžeme si už ale byť istí, že sú určite práve tri. Vaša funkcia `count_mushrooms` už teda môže ukončiť výpočet a vrátiť na výstupe hodnotu 3.

Obmedzenia

- $2 \leq n \leq 20\,000$

Hodnotenie

Ak čo len v jednom teste porušíte vyššie popísané obmedzenia (pre maximálny počet spustení mašiny a maximálny celkový počet testovaných húb), za celú úlohu dostanete 0 bodov.

Ak čo len v jednom teste vaša funkcia `count_mushrooms` vráti nesprávny výstup, za celú úlohu dostanete 0 bodov.

V ostatných prípadoch sa vaše body za túto úlohu vypočítajú nasledovne: Pre každý test sa pozrieme na to, koľkokrát ste použili mašinu. Maximum z týchto počtov označíme Q . Výpočet vášho počtu bodov z hodnoty Q udáva nasledujúca tabuľka:

| Rozsah hodnôt | Skóre |
|----------------------------|---|
| $20\,000 < Q$ | 0 |
| $10\,010 < Q \leq 20\,000$ | 10 |
| $1000 < Q \leq 10\,010$ | 25 |
| $250 < Q \leq 1000$ | $25 + \left(\frac{1000-Q}{750}\right)^2 \cdot 75$ |
| $Q \leq 250$ | 100 |

V niektorých testoch sa bude grader správať adaptívne. To znamená, že na začiatku takéhoto testu neexistuje žiadna konkrétna postupnosť typov húb. Grader si pri každom volaní funkcie `use_machine` môže vybrať, čo vám odpovie, na základe všetkých volaní `use_machine`, ktoré ste dovtedy spravili. V týchto testoch je zaručené, že odpovede gradera budú konzistentné: v ľubovoľnom okamihu bude platiť, že existuje aspoň jedna postupnosť typov húb, ktorá zodpovedá všetkým odpovediam, ktoré vám grader v tomto teste dal.

Ukážkový grader

Ukážkový grader očakáva na vstupe postupnosť čísel $s[0], \dots, s[n-1]$, pričom $s[i] = 0$ znamená, že huba i je typu A, zatiaľ čo $s[i] = 1$ znamená, že huba i je typu B. Tento vstup má byť formátovaný nasledovne:

- line 1: n
- line 2: $s[0] \ s[1] \ \dots \ s[n-1]$

Výstup ukážkového gradera:

- line 1: návratová hodnota vašej funkcie `count_mushrooms`.
- line 2: počet volaní `use_machine`, ktoré spravila.

Podotýkame ešte, že ukážkový grader nie je adaptívny.

