

Kefa na kočky

Noro berie súťaž o zber najvyššej živočíšnej hmotnosti seriózne. V Kluži už ulovil pár tuctov mačiek, ktoré skladuje na izbe v kúpeľni a ktoré po nociach kefuje.

Na toto hobby má Noro pripravenú kefu s kapacitou na chlpy c . V kúpeľni väzní n mačiek, ktoré si láskyplne pomenoval 0 až $n - 1$, a ktoré musí každú noc vykefovať v tomto poradí. Každá cica i obsahuje $v[i]$ voľných chlpy a trvá $a[i]$ sekúnd kým cez ňu Noro pretiahne kefu. Mačky neznášajú keď ich Noro náhle prestane kefovať, takže pretáhovanie kefy musí vždy prebehnúť úplne.

Po prejdení niekoľkých mačiek sa môže kapacita kefy naplniť a upchať chlpmi. V tom prípade kefa prestane zbierať chlpy a zvyšné množstvo chlpy na mačke zostane. Zakaždým, keď sa to stane, je potrebné kefu vyčistiť, čo je otravné a trvá b sekúnd. Ak sa kefa naplní počas kefovania mačky i , Noro musí kefu pretiahnuť až do konca, vyčistiť kefu a potom mačku prejsť znova (alebo aj viackrát), aby pozbieral zvyšné chlpy. Napríklad ak cez mačku i musí prejsť 3-krát, aby pozbieral všetky chlpy, bude to trvať $a[i] + b + a[i] + b + a[i]$ sekúnd. **Po vykefovaní všetkých mačiek musí byť kefa čistá, teda ju bude treba ešte raz vyprázdniť.**

Norka dnes na obede napdalo, že niekedy môže byť vlastne efektívnejšie vyčistiť kefu aj predtým, než sa úplne naplní – no nie je si istý, aká stratégia je najlepšia. Rád by sa pred súťažou čo najviac vyspal, preto žiada o pomoc teba (ak si Noro, pomôž si sám).

Na základe množstva chlpy na jednotlivých mačkách a počtu sekúnd potrebných na ich pretiahnutie, kapacity kefy a času na jej vyprázdnenie, nájdi najlepší spôsob, ako Noro vykefuje mačky za minimálny čas.

Implementačné detaily

Implementuj nasledujúcu funkciu (Rumunom v názve chýba e , ale už sa to nedá zmeniť):

```
long long mow(int n, int c, int b, std::vector<int>& a, std::vector<int>& v);
```

- n : počet mačiek
- c : celková kapacita kefy
- b : počet sekúnd potrebných na vyčistenie kefy
- a : vektor dĺžky n popisujúci čas potrebný na jeden prechod kefy cez jednotlivé mačky
- v : vektor dĺžky n udávajúci množstvo voľných chlpy na jednotlivých mačkách
- Funkcia by mala vrátiť jediné celé číslo – minimálny čas na vykefovanie všetkých mačiek.
- Procedúra bude volaná práve raz pre každý testovací prípad.

Obmedzenia

- $1 \leq n \leq 200\,000$
- $1 \leq a[i] \leq 10^9$ (pre každé i , $0 \leq i < n$)
- $1 \leq v[i] \leq 10^9$ (pre každé i , $0 \leq i < n$)
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- Je zaručené, že správny výsledok bude najviac 10^{18}

Podúlohy

1. (9 bodov) Všetky hodnoty $(n, b, c, a[i], v[i])$ budú najviac 200
2. (16 bodov) $n, c \leq 5000$ a $v[i] \leq 5000$ pre všetky $0 \leq i < n$
3. (36 bodov) $c \leq 200\,000$
4. (17 bodov) $a[0] = a[1] = \dots = a[n-1]$
5. (22 bodov) Bez ďalších obmedzení

Príklady

Príklad 1

```
mow(3, 5, 2, {2, 10, 3}, {2, 4, 6})
```

V tomto príklade sú 3 mačky, kefa má kapacitu 5 a jej vyčistenie trvá 2 sekundy.

Noro prebehne prvú mačku za 2 sekundy, v kefe budú 2 chlpy. Potom sa rozhodne kefu vyčistiť za 2 sekundy. Na prvej mačke strávi spolu 4 sekundy.

Potom pokefuje druhú mačku – všetky 4 chlpy, nezaplní kefu, nevyprázdňuje. To trvá 10 sekúnd.

Na tretej mačke sa mu kefa naplní hneď po prvom chlpe, dokončí prechod kefy, vyčistí kefu, potom mačku znovu prejde a kefu znova vyčistí. Nezabudnite, že kefa musí byť na konci čistá. Celkovo to bude trvať $3 + 2 + 3 + 2 = 10$ sekúnd.

Celkovo strávi $4 + 10 + 10 = 24$ sekúnd.

Príklad 2

```
mow(4, 10, 4, {1, 2, 1, 4}, {3, 2, 6, 7})
```

Optimálna stratégia je vykefovať prvé 3 mačky bez vyprázdňovania, potom vyčistiť kefu, pokefovať posledné 2 mačky a nakoniec vyprázdniť kefu. Výsledný čas bude:

$$a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17 \text{ sekúnd.}$$

Hodnotič príkladov

Hodnotič číta vstup v tomto formáte:

- riadok 1: n c b
- riadok 2: $a[0]$ $a[1]$ \dots $a[n - 1]$
- riadok 3: $v[0]$ $v[1]$ \dots $v[n - 1]$

a vypíše výsledok volania `mow` s príslušnými parametrami.