Hídépítés

Időlimit: 3 s *Memórialimit: 128 MB*

Egy széles folyón n darab oszlop áll ki a vízből. Az i. oszlop magassága h_i. Az oszlopok egy egyenes vonal mentén helyezkednek el a két part között. Ezeknek az oszlopoknak a felhasználásával egy hidat szeretnénk építeni. Ennek érdekében kiválasztjuk oszlopok egy részhalmazát, és a tetejüket hídrészekkel kötjük össze. A részhalmaznak tartalmaznia kell az első és az utolsó oszlopot.

Az i. és j. oszlop közötti hídrész építésének költsége $(h_i - h_j)^2$. A nem használt oszlopokat emellett el kell tüntetni. Az i. oszlop eltüntetésének költsége pedig w_i, ami negatív is lehet (lehet, hogy valaki hajlandó fizetni az oszlop eltávolításáért).

Írj programot, amely meghatározza, hogy mekkora a lehető legkisebb költsége az első és utolsó oszlopot összekötő híd építésének!

Bemenet

A bemenet első sora az oszlopok n számát tartalmazza. A második sorban az oszlopok h_i magasságai vannak i szerint növekvő sorrendben, szóközökkel elválasztva. A harmadik sor ugyanebben a sorrendben az eltávolítás w_i költségeit tartalmazza. Minden magasság és költség értéke egész szám.

Kimenet

A kimenetre a híd építésének minimális költségét kell kiírni. Megjegyezzük, hogy ez negatív is lehet.

Korlátok

- $2 \leq n \leq 10^5$
- $0 \leq h_i \leq 10^6$
- $0 \leq |w_i| \leq 10^6$

1. tesztcsoport (30 pont)

- $n \leq 1\,000$

2. tesztcsoport (30 pont)

- Az optimális megoldás az első és utolsó oszlop mellett legfeljebb 2 további oszlop felhasználásával érhető el.
- $|w_i| \leq 20$

3. tesztcsoport (40 pont)

- További korlátok nincsenek.

Példa

Bemenet

```
6
3 8 7 1 6 6
0 -1 9 1 2 0
```

Kimenet

```
17
```

Tükrös felosztások

Időlimit: 10 s *Memórialimit: 128 MB*

Egy s szó *partíció*ja olyan $a_1, a_2, a_3, \ldots, a_d$ sorozat, ahol a sorozat a_i elemei az s szónak nem üres és nem átfedő összefüggő szövegrészei, és az egymásután írásuk az s szót adja: $s = a_1 + a_2 + a_3 + \ldots + a_d$. Egy ilyen partíció hosszán a sorozat elemeinek d számát értjük.

Egy partíciót megadhatunk egy olyan szöveggel, hogy a partíció elemeit zárójelbe téve egymás után írjuk. A `"decode"` szó lehetséges partíciói például a következői: `(d)(ec)(ode)`, `(d)(e)(c)(od)(e)`, `(decod)(e)`, `(decode)`, `(de)(code)` (de még más is lehet).

Egy partíció *tükrös*, ha elemeit atomi egységnek tekintve, ezen egységekből képzett szó tükörszó (azaz balról jobbra és jobbról balra olvasva megegyeznek). Például a `"decode"` szó összes tükrös partíciója a `(de)(co)(de)` és `(decode)`. Ez utóbbi arra is rávilágít, hogy minden szónak van egy triviális partíciója, ami egy elemből áll.

Írj programot, amely kiszámítja egy adott szó leghosszabb partíciójának hosszát. Figyelmeztetünk, hogy a hossz a partíció elemeinek számát jelenti.

Bemenet

A bemenet első sorában a tesztesetek t száma van. A következő t sor mindegyike egy-egy s szót tartalmaz, ami csak az angol ábécé kisbetűiből áll. A szavak szóközt nem tartalmaznak.

Kimenet

A kimenetre t sort kell kiírni: minden tesztesetre a megoldás értékét, azaz adott szó leghosszabb partíciójának hosszát.

Korlátok

Jelölje n az s szó hosszát!

- $1 \leq t \leq 10$
- $1 \leq n \leq 10^6$

1. tesztcsoport (30 pont)

- $n \leq 30$

2. tesztcsoport (20 pont)

- $n \leq 300$

3. tesztcsoport (25 pont)

- $n \leq 10\,000$

4. tesztcsoport (25 pont)

- További korlátok nincsenek.

Példa

Bemenet	Kimenet
4	3
bonobo	5
deleted	7
racecar	1
racecars	

Hajsza

Időlimit: 4 s   *Memórialimit: 512 MB*

Tom, a macska szokása szerint kergeti Jerryt, az egeret. Jerry azáltal próbál előnyre szert tenni, hogy galambok közé menekül, ahol Tom lassabban tudja őt követni. Jerry a ljubljanai Central Parkba menekült, ahol n szobor van $1 \dots n$ számozva, amiket $n-1$ egymást nem metsző, közvetlen ösvény köt össze. Ezeken keresztül bármelyik szobortól bármelyik szoborhoz el lehet jutni. Az i szobornál p_i számú galamb van. Jerrynek v számú kenyérdarab van a zsebében. Ha egy szobornál leejt egy kenyérdarabot, akkor minden szomszédos szobortól az összes galamb odagyűlik. Ennek eredményeképpen mind az adott, mind a szomszédos szobroknál megváltozik a galambok száma.

Kenyérdarab leejtése esetén az események a következőképpen követik egymást: Jerry először megérkezik az i. szoborhoz, ahol p_i galamb van. Leejt egy kenyérdarabot, majd elindul egy szomszédos szobor felé. A galambok a szomszédos szobroktól az i.-hez mennek, mielőtt Jerry elérné a következő szobrot (azaz nem kell számolni azokkal a galambokkal, amelyek a két szobor között mozognak).

Jerry bármelyik szobornál beléphet a parkba, ahol bármely útvonalon haladhat úgy, hogy ugyanazon az ösvényen nem mehet kétszer, végül bárhol elhagyhatja a parkot. Miután Jerry elhagyta a parkot, Tom ugyanazon az útvonal halad végig. v kenyérdarab leejtésével Jerry maximalizálni akarja azon galambok számának különbségét, amelyekkel az útvonal bejárása során a szobroknál találkoznak. Még egyszer hangsúlyozzuk, hogy csak azok a galambok számítanak, amik a szobornál jelen vannak közvetlenül azelőtt, hogy Jerry odalépne. Lásd még a példához tartozó magyarázatot.

Bemenet

A bemenet első sora a szobrok n számát és a kenyérdarabok v számát tartalmazza. A második sor n darab, szóközzel elválasztott egész számot tartalmaz, az i. szám az i. szobornál lévő galambok p_i száma $(p_1 \dots p_n)$. A további $n-1$ sor mindegyike két számot, a_i-t és b_i-t tartalmazza, ami azt jelenti, hogy az a_i. és b_i. szobor között van ösvény.

Kimenet

A kimenetre egyetlen számot, azt a legnagyobb különbséget kell kiírni, ami azon galambok számának különbsége, amelyekkel az útvonal bejárása során Jerry és Tom a szobroknál találkoznak.

Korlátok

- $1 \le n \le 10^5$
- $0 \le v \le 100$
- $0 \le p_i \le 10^9$

1. tesztcsoport (20 pont)

- $1 \le n \le 10$

2. tesztcsoport (20 pont)

- $1 \leq n \leq 1000$

3. tesztcsoport (30 pont)

- Egy optimális útvonal az 1. szobornál kezdődik.

4. tesztcsoport (30 pont)

- További korlátok nincsenek.

Példa

Bemenet

```
12 2
2 3 3 8 1 5 6 7 8 3 5 4
2 1
2 7
3 4
4 7
7 6
5 6
6 8
6 9
7 10
10 11
10 12
```

Kimenet

```
36
```

Megjegyzés

Egy lehetséges megoldás a következő. Jerry a 6. szobornál lép a parkba. Ott 5 galamb van. Leejt egy kenyérdarabot. Ennek hatására $p_6 = 27$ és $p_5 = p_7 = p_8 = p_9 = 0$ lesz. Ezt követően a 7. szoborhoz szalad, ahol 0 galambot talál. Leejti a második kenyérdarabot. Ekkor $p_7 = 41$ és $p_2 = p_4 = p_6 = p_{10} = 0$ lesz. Majd kimegy a parkból. Útja során $5 + 0 = 5$ galambbal találkozott. Ezután Tom ugyanezt az útvonalat járja be, de ő $p_6 + p_7 = 0 + 41 = 41$ galambbal találkozik. A különbség kettejük között $41 - 5 = 36$, ami optimális.