Stavba mostů (Building Bridges)

Časový limit: 3 s *Paměťový limit: 128 MB*

Z vody v široké řece ční nad hladinu n pilířů o nějakých, potenciálně různých, výškách. Pilíře jsou umístěné v rovné řadě od jednoho břehu ke druhému. Chtěli bychom postavit most, který na nich bude stát. Abychom toho dosáhli, vybereme podmnožinu pilířů a jejich vršky spojíme, tím vzniknou části mostu. Vybraná podmnožina musí obsahovat první a poslední pilíř.

Protože se chceme vyhnout nerovnostem, vybudování části mostu mezi pilíři i a j stojí $(h_i - h_j)^2$, kde h_i je výška pilíře i. Navíc musíme odstranit všechny pilíře, které nejsou součástí mostu, protože blokují říční dopravu. Cena odstranění i-tého pilíře je w_i. Tato cena může být i záporná – zainteresované strany jsou ochotné zaplatit vám, abyste se určitých pilířů zbavili. Všechny výšky h_i i ceny w_i jsou celá čísla.

Jaká je nejmenší možná cena vybudování mostu, který spojuje první a poslední pilíř?

Vstup

První řádek obsahuje počet pilířů n. Druhý řádek obsahuje popořadě výšky h_i, oddělené mezerami. Třetí řádek pak obsahuje ceny w_i za odstranění pilířů, a to ve stejném pořadí.

Výstup

Vypište nejmenší cenu vybudování mostu. Uvědomte si, že může být záporná.

Omezení

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

Podúloha 1 (30 body)

- $n \leq 1\,000$

Podúloha 2 (30 body)

- Optimální řešení zahrnuje nejvýše 2 dodatečné pilíře (tj. první, poslední a maximálně 2 další).
- $|w_i| \leq 20$

Podúloha 3 (40 body)

- bez dalších omezení

Příklad

Vstup

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

Výstup

```
17
```

Palindromická dělení (Palindromic Partitions)

Časový limit: 10 s *Paměťový limit: 128 MB*

Dělení řetězce s definujeme jako množinu jednoho nebo více nepřekrývajících se neprázdných podřetězců řetězce s (označme je $a_1, a_2, a_3, \ldots, a_d$) takových, že s vznikne jejich zřetězením: $s = a_1 + a_2 + a_3 + \ldots + a_d$. Tyto podřetězce nazveme „*kousky*“ a zadefinujeme si *délku* dělení jako počet kousků, d.

Dělení řetězce můžeme reprezentovat tím, že každý kousek napíšeme do závorek. Například řetězec `"decode"` můžeme rozdělit jako `(d)(ec)(ode)`, `(d)(e)(c)(od)(e)`, `(decod)(e)`, `(decode)`, `(de)(code)`, nebo několika dalšími způsoby.

Dělení označíme za *palindromické*, pokud jeho kousky tvoří palindrom, kousky v tomto případě považujeme za nedělitelné jednotky. Například jedinými palindromickými děleními slova `"decode"` jsou `(de)(co)(de)` a `(decode)`. Z příkladu si také můžeme všimnout, že každé slovo má triviální palindromické dělení délky jedna.

Vaším úkolem je spočítat maximální možný počet kousků palindromického dělení.

Vstup

Vstup začíná řádkem obsahujícím počet testovaných případů t. Následujících t řádků popisuje jednotlivé testované případy – pokaždé jedno slovo s, které obsahuje pouze malé znaky anglické abecedy. Vstup neobsahuje žádné mezery.

Výstup

Pro každý testovaný případ vypište jedno číslo udávající délku nejdelšího palindromického dělení daného slova s.

Omezení

Počet znaků vstupního řetězce s označíme n.

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

Podúloha 1 (15 bodů)

- $n \leq 30$

Podúloha 2 (20 bodů)

- $n \leq 300$

Podúloha 3 (25 bodů)

- $n \leq 10\,000$

Podúloha 4 (40 bodů)

- bez dalších omezení

Příklad

Vstup	Výstup
`4`	`3`
`bonobo`	`5`
`deleted`	`7`
`racecar`	`1`
`racecars`	

Honička (Chase)

Časový limit: 4 s *Paměťový limit: 512 MB*

Kocour Tom už zase honí myšáka Jerryho! Jerry se snaží získat náskok tím, že vběhne do hejna holubů, ve kterém je pro Toma těžší jej sledovat. Ke svému štěstí Jerry dorazil do ústředního parku v Lublani. V parku se nachází n soch očíslovaných $1 \dots n$ a $n-1$ nekřížících se pěšinek spojujících je takovým způsobem, že je možné se ke každé soše dostat od jakékoli jiné chůzí po pěšinkách. Kolem každé sochy i je těsně shromážděno p_i holubů. Jerry má v kapse v chlebových drobků. Pokud u sochy (u které se zrovna nachází) upustí drobek, slétnou se sem co nejdříve nazobat holubi ze všech sousedních soch. Důsledkem toho se aktuální počet holubů p u této a sousedních soch změní.

Všechno se odehraje v následujícím pořadí: Nejprve Jerry dorazí k soše i a potká p_i holubů. Poté odhodí drobek a opustí sochu. Holubi ze všech sousedních soch přeletí k soše i ještě před Jerryho příchodem k další soše (takže se nepočítají k počtu holubů, které potkal).

Jerry může vběhnout do parku u libovolné sochy, běžet po pěšinkách (jen nesmí nikdy běžet po stejné pěšince dvakrát) a nakonec vyběhnout u libovolné sochy. Až Jerry opustí park, vběhne do parku Tom a poběží po té samé cestě. Jerry se snaží maximalizovat rozdíl v počtu holubů, které na cestě potká Tom a které potká Jerry, tím, že odhodí až v drobků. Všimněte si, že Jerrymu se jako potkaní počítají jen ti holubi, kteří jsou u sochy v okamžiku jeho příchodu. Pro další vysvětlení si prostudujte komentář k příkladu.

Vstup

První řádek se skládá z počtu soch n a počtu chlebových drobků v. Druhý řádek obsahuje n celých čísel oddělených mezerami, $p_1 \dots p_n$. Následujících $n-1$ řádků popisuje pěšinky pomocí dvojice čísel a_i a b_i, což značí, že pěšinka vede mezi sochami a_i a b_i.

Výstup

Vypište pouze jedno číslo – maximální možný rozdíl mezi počtem holubů, které potká Tom a které Jerry.

Omezení

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

Podúloha 1 (20 bodů)

- $1 \le n \le 10$

Podúloha 2 (20 bodů)

- $1 \le n \le 1000$

Podúloha 3 (30 bodů)

- Optimální cesta začíná u sochy číslo 1.

Podúloha 4 (30 bodů)

- bez dalších omezení

Příklad

Vstup

```
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
```

Výstup

```
36
```

Poznámka

Jedno z možných řešení je následující: Jerry vstoupí do parku u sochy 6. U ní potká 5 holubů a odhodí drobek. Nyní platí, že p_6 je rovno 27 a $p_5 = p_7 = p_8 = p_9 = 0$. Následně přiběhne k soše 7 a potká 0 holubů. Opět upustí drobek. V p_7 je nyní 41 a $p_2 = p_4 = p_6 = p_{10} = 0$. Jerry opustí park. Dohromady potkal $5 + 0 = 5$ holubů. Tom jej následuje po stejné cestě, ale potká $p_6 + p_7 = 0 + 41 = 41$ holubů. Rozdíl je $41 - 5 = 36$.