Budujemy mosty

Limit czasu: 3 s   *Limit pamięci: 128 MB*

Z szerokiej rzeki wystaje n kolumn o być może różnych wysokościach. Są one usytuowane w linii prostej pomiędzy dwoma brzegami rzeki. Chcemy zbudować oszałamiający most, który opiera się na kolumnach. Aby to osiągnąć, wybieramy podzbiór kolumn i budujemy most oparty o szczyty kolejno wybranych kolumn. Pierwsza i ostatnia kolumna musi być wybrana do tego podzbioru.

Koszt zbudowania odcinka mostu między kolejnymi wybranymi kolumnami i oraz j wynosi $(h_i - h_j)^2$ (gdzie h_i oznacza wysokość i-tej kolumny). Ponadto musimy zapłacić za usunięcie niewybranych kolumn (na których most nie jest oparty), ponieważ zaburzają one bieg rzeki. Koszt usunięcia i-tej kolumny jest równy w_i. Koszt ten może być ujemny – niektórzy mogą być skłonni zapłacić, aby pozbyć się określonych kolumn. Wszystkie wysokości h_i oraz koszty w_i są liczbami całkowitymi.

Jaki jest minimalny koszt zbudowania mostu, który połączy pierwszą i ostatnią kolumnę?

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę naturalną n. Drugi wiersz wejścia zawiera n liczb całkowitych h_i, pooddzielanych pojedynczymi odstępami. Są to wysokości kolumn w kolejności ich występowania na rzece. Trzeci wiersz zawiera liczby w_i, w tej samej kolejności i w tym samym formacie. Oznaczają one koszt usunięcia kolejnych kolumn.

Wyjście

Wypisz minimalny koszt wybudowania mostu. Zauważ, że może on być ujemny.

Ograniczenia

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

Podzadanie 1 (30 punktów)

- $n \le 1\,000$

Podzadanie 2 (30 punktów)

- optymalne rozwiązanie używa co najwyżej dwóch dodatkowych kolumn (nie licząc pierwszej i ostatniej)
- $|w_i| \le 20$

Podzadanie 3 (40 punktów)

- brak dodatkowych ograniczeń

Przykład

Wejście

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

Wyjście

```
17
```

Palindromiczny podział

Limit czasu: 10 s *Limit pamięci: 128 MB*

Podział napisu s jest sekwencją złożoną z jednego lub więcej nienakładających się, niepustych spójnych podciągów s (nazwijmy je $a_1, a_2, a_3, \ldots, a_d$), takich że s jest ich konkatenacją: $s = a_1 + a_2 + a_3 + \ldots + a_d$.

Te spójne podciągi nazywamy kawałkami, a długością podziału nazwiemy liczbę jego kawałków d.

Aby wygodnie reprezentować podział napisu, możemy umieścić kolejne jego kawałki w nawiasach. Na przykład: napis „`decode`" może być podzielony jako `(d)(ec)(ode)`, `(d)(e)(c)(od)(e)`, `(decod)(e)`, `(decode)`, `(de)(code)` oraz na wiele innych sposobów.

Podział nazwiemy *palindromicznym* jeśli jego kawałki tworzą palindrom, gdy rozważamy każdy kawałek jako pojedynczy obiekt. Na przykład: jedyne palindromiczne podziały napisu „`decode`" to: `(de)(co)(de)` oraz `(decode)`. To pokazuje (między innymi), że każdy napis ma trywialny palindromiczny podział długości jeden (czyli składający się z jednego kawałka).

Twoim zadaniem jest obliczyć największą możliwą liczbę kawałków w palindromicznym podziale zadanego napisu.

Wejście

Wejście rozpoczyna się liczbą zestawów testowych t w pierwszym wierszu. Kolejnych t wierszy zawiera pojedyncze zestawy testowe składające się zawsze z jednego napisu s, złożonego jedynie z małych liter alfabetu angielskiego. W napisach nie ma żadnych odstępów.

Wyjście

Dla każdego zestawu testowego wypisz jedną liczbę: długość (liczbę kawałków) najdłuższego palindromicznego podziału napisu s z wejścia.

Ograniczenia

Oznaczmy długość napisu s jako n.

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

Podzadanie 1 (15 punktów)

- $n \leq 30$

Podzadanie 2 (20 punktów)

- $n \leq 300$

Podzadanie 3 (25 punktów)

- $n \leq 10\,000$

Podzadanie 4 (40 punktów)

- brak dodatkowych ograniczeń

Przykład

Wejście	**Wyjście**
`4`	`3`
`bonobo`	`5`
`deleted`	`7`
`racecar`	`1`
`racecars`	

Pościg Mariusza

Limit czasu: 4 s *Limit pamięci: 512 MB*

Słoń Mariusz (znów) chce dopaść myszkę Jerry. Nadzieją Jerry'ego jest wykorzystanie stad gołębi, które mogą opóźnić wielkiego, wspaniałego słonia Mariusza.

Jerry, uciekając przed Mariuszem, wbiegnie za chwilę do parku w Lublanie. Park zawiera n pomników, ponumerowanych $1, 2, \ldots, n$, oraz $n - 1$ dróżek łączących pomniki w taki sposób, że z każdego pomnika da się dojść do każdego innego pomnika. Niech p_i oznacza liczbę gołębi, które w danej chwili obsiadają (i ozdabiają) i-ty pomnik. Początkowe wartości p_i są dane na wejściu.

Jerry ma v okruszków w swoich kieszeniach. Gdy rzuca okruszek przy mijanym pomniku, gołębie z sąsiednich pomników po chwili przylecą do tego pomnika, by się posilić. Zauważ, że zmieni to liczby gołębi p_x przy tym i sąsiednich pomnikach.

Wszystko dzieje się w następującej kolejności: Najpierw Jerry pojawia się przy pomniku i i spotyka p_i gołębi (ta liczba może różnić się od liczby p_i z wejścia, bo być może liczba gołębi się zmieniła). Potem może rzucić okruszek. W końcu oddala się od pomnika. Jeśli Jerry rzucił okruszek, gołębie z sąsiednich pomników zrywają się do lotu i po chwili pojawiają się przy i-tym pomniku (lecą one bardzo wysoko, więc Jerry nie spotyka ich w tym momencie).

Jerry może wejść do parku przy dowolnym pomniku, użyć niektórych dróżek (nie używając żadnej więcej niż raz) i w końcu opuścić park przy dowolnym pomniku. Gdy Jerry opuści park, słoń Mariusz wkroczy do parku i podąży dokładnie tą samą trasą. Używając co najwyżej v okruszków, Jerry chce zmaksymalizować różnicę między liczbą gołębi, które spotka Mariusz, a liczbą gołębi, które spotka on sam. Zauważ, że Jerry spotyka tylko te gołębie, które są przy pomniku, gdy przychodzi on do tego pomnika (nie spotyka w tym momencie gołębi, które przylecą po okruszek). Przeczytaj wyjaśnienie do przykładu poniżej, by rozjaśnić ewentualne wątpliwości.

Wejście

Pierwszy wiersz wejścia zawiera liczbę pomników n oraz liczbę okruszków v. Drugi wiersz zawiera n liczb całkowitych oddzielonych spacjami $p_1, \ldots, p_n$ – początkowe liczby gołębi przy pomnikach. Każdy z kolejnych $n - 1$ wierszy zawiera dwie liczby a_i i b_i, opisujące dróżkę między pomnikami a_i i b_i.

Wyjście

Wypisz jedną liczbę, oznaczającą największą możliwą różnicę między liczbą gołębi, które spotka Mariusz, oraz liczbą gołębi, które spotka drobna myszka Jerry.

Ograniczenia

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

Podzadanie 1 (20 punktów)

- $1 \le n \le 10$

Podzadanie 2 (20 punktów)

- $1 \le n \le 1\,000$

Podzadanie 3 (30 punktów)

- optymalna trasa Jerry'ego zaczyna się przy pomniku numer 1

Podzadanie 4 (30 punktów)

- brak dodatkowych ograniczeń

Przykład

Wejście

```
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
```

Wyjście

```
36
```

Komentarz

Opiszemy jedną możliwą optymalną strategię Jerry'ego. Wkracza do parku przy pomniku 6. Spotyka 5 gołębi. Rzuca okruszek. p_6 zmienia się w 27 oraz $p_5 = p_7 = p_8 = p_9 = 0$. Wtedy przechodzi do pomnika 7 i spotyka 0 gołębi. Rzuca drugi okruszek. p_7 zmienia się w 41 oraz $p_2 = p_4 = p_6 = p_{10} = 0$. Jerry opuszcza park. Spotkał on $5 + 0 = 5$ gołębi. Mariusz podąża za myszą tą samą trasą i napotyka $p_6 + p_7 = 0 + 41 = 41$ gołębi. Różnica wynosi $41 - 5 = 36$.