



Building Bridges

Time Limit: 3 s Memory Limit: 128 MB

Pe un rau sunt instalati N piloni de posibil diferite inaltimei care ies din apa. Ei sunt asezati pe o linie de la un mal la celalalt. Tebuie construit un pod care sa foloseasca pilonii ca suport. Pentru acesta se alege o submultime de piloni (care sa includa primul si ultimul pilon, cei doi fiind aflati langa cele doua maluri) si sa le conectam partile de sus prin sectiuni ale podului.

Costul construirii unei sectiuni de pod intre pilonii i si j este $(h(i) - h(j))^2$ unde h_i este inaltimea pilonului i . In plus, trebuie eliminati toti pilonii care nu sunt inclusi in pod, deoarece vor impiedica traficul pe rau sau rostogolirile pe pod. Costul eliminarii celui de-al i -lea pilon este w_i . Acest cost poate fi chiar negativ—cateva grupuri de interese vor sa plateasca pentru eliminarea unor piloni. Toate inaltimele h_i si costurile w_i sunt intregi.

Care este costul minim posibil al construirii podului care conecteaza primul si ultimul pilon?

Input

Prima linie contine numarul de piloni, N . A doua linie contine in ordine inaltimele pilonilor h_i , separate prin cate un spatiu. A treia linie contine w_i in aceeasi ordine, costul eliminarii pilonilor.

Output

Sa se afiseze costul minim necesar pentru construirea podului. Atentie la faptul ca acest cost poate fi negativ.

Restrictii

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

Subtask 1 (30 de puncte)

- $N \leq 1\,000$

Subtask 2 (30 de puncte)

- solutia optima contine cel mult 2 piloni suplimentari (in afara de primul si ultimul)
- $|w_i| \leq 20$

Subtask 3 (40 de puncte)

- fara alte restrictii

Exemplu



Input

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

Output

```
17
```



Palindromic Partitions

Time Limit: 10 s Memory Limit: 128 MB

O *partitie* a unui string S este un set de una sau mai multe subsecvente nevide a lui S care nu se intersecteaza (sa le numim $a_1, a_2, a_3, \dots, a_d$), astfel incat S este concatenarea acestor subsecvente: $S = a_1 + a_2 + a_3 + \dots + a_d$. Numim *lungimea* unei astfel de partitii ca fiind numarul de subsecvente din partitie, d .

Putem reprezenta partitia unui string prin incadrarea fiecărei subsecvente in paranteze. De exemplu, stringul "decode" poate sa fie partitionat ca (d)(ec)(ode) sau (d)(e)(c)(od)(e) sau (decod)(e) sau (decode) sau (de)(code) sau in multe alte moduri.

O partitie este *palindromica* daca subsecventele sale formeaza un sir palindrom, considerand fiecare subsecventa ca un element atomic. De exemplu, singurele partitii atomice ale cuvântului "decode" sunt (de)(co)(de) si (decode). Acest lucru ilustreaza faptul ca fiecare cuvânt are o partitie palindromica triviala de lungime unu.

Sarcina voastra este sa calculati lungimea maxima a unei partitii palindromice, pentru un cuvânt dat.

Input

Pe prima linie se afla T , numarul de teste. Urmatoarele T linii descriu individual cate un test, reprezentate printr-un singur cuvânt S ce contine caractere mici ale alfabetului Englez. Nu exista spatii in input.

Output

Pentru fiecare test, afisati un singur numar: lungimea celei mai lungi partitii palindromice a cuvântului S din input.

Restrictii

Sa notam lungimea stringului S cu N .

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^6$

Subtask 1 (15 puncte)

- $N \leq 30$

Subtask 2 (20 de puncte)

- $N \leq 300$

Subtask 3 (25 de puncte)

- $N \leq 10\,000$



Subtask 4 (40 de puncte)

- Fara restrictii suplimentare.

Example

Input	Output
4	3
bonobo	5
deleted	7
racecar	1
racecars	



Chase

Time Limit: 4 s Memory Limit: 512 MB

Motanut Tom il vaneaza si este popit de soricelul Jerry din nou! Jerry incearca ca castige ceva avantaj alergand prin stoluri de porumbei, unde este mai greu pentru Tom sa-l urmareasca. Jerry a ajuns in Parcul Central din Ljubljana. Parcul are N statui, numerotate cu numere din intervalul $1 \dots N$, si $N - 1$ alei care nu se intersecteaza le leaga in asa fel incat este posibil sa se ajunga de la o statuie la oricare alta mergand pe aceste alei. In jurul fiecarei statui i se afla adunati p_i porumbei. Jerry are V firmituri in buzunar. Daca el arunca o firmitura la statuia in care se afla, toti porumbeii de pe statuile vecine vor zbura imediat catre aceasta statuie pentru a manca firmiturile de paine. Prin urmare, numarul de porumbei p din statuia curenta si statuile vecine se modifica.

Totul se intampla in felul urmator: Mai intai, Jerry ajunge la statuia i si intalneste p_i porumbei. Dupa, acesta arunca firmiturile. Mai apoi, acesta pleaca de la statuie. In final, porumbeii din statuile vecine zboara catre statuia i inainte ca Jerry sa ajunga la urmatoarea statuie (deci acesti porumbei nu se contorizeaza la numarul de porumbei pe care soricelul ii intalneste).

Jerry poate sa intre in parc la orice statuie, parcurge cateva alei (dar niciodata nu foloseste o alee de doua ori), iar in final paraseste parcul pe oriunde vrea. Dupa ce Jerry paraseste parcul, Tom intra si parcurge exact acelasi drum. Prin aruncarea a cel mult V firmituri, Jerry vrea sa maximizeze diferenta dintre numarul de porumbei pe care Tom ii va intalni si numarul de porumbei pe care el ii va intalni. De remarcat faptul ca doar porumbeii care sunt prezenti la o statuie fix inainte ca Jerry sa ajunga se numara la numarul total de porumbei pe care acesta ii intalneste. Verificati explicatiile exemplului pentru mai multe detalii.

Input

Prima linie contine numarul de statui N si numarul de firmituri V . A doua linie contine N intregi separati prin cate un spatiu, $p_1 \dots p_n$. Urmatoarele $N - 1$ linii descriu aleile prin perechi de numere a_i si b_i , insemnand ca exista o cale de la statuia a_i la statuia b_i .

Output

Afisati un singur numar, diferenta maxima dintre numarul de porumbei pe care Tom ii intalneste si numarul de porumbei pe care Jerry ii intalneste.

Restrictii

- $1 \leq N \leq 10^5$
- $0 \leq V \leq 100$
- $0 \leq p_i \leq 10^9$

Subtask 1 (20 de puncte)

- $1 \leq N \leq 10$



Subtask 2 (20 de puncte)

- $1 \leq N \leq 1000$

Subtask 3 (30 de puncte)

- Un drum optim incepe de la statuia 1.

Subtask 4 (30 de puncte)

- Fara restrictii suplimentare.

Example

Input

```
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
```

Output

```
36
```

Explicatii

O posibila solutie este urmatoarea. Jerry intra in parc la statuia 6. Acolo intalneste 5 porumbei. Arunca o firmitura la aceasta statuie. p_6 este acum 27 si $p_5 = p_7 = p_8 = p_9 = 0$. Dupa alearga catre statuia 7 si intalneste 0 porumbei. Arunca cea de a doua firmitura. p_7 este acum 41 si $p_2 = p_4 = p_6 = p_{10} = 0$. Dupa iese din parc si a intalnit $5 + 0 = 5$ porumbei. Tom urmeaza fix acelasi drum si intalneste $p_6 + p_7 = 0 + 41 = 41$ porumbei. Diferenta este $41 - 5 = 36$.