Problem A. Pęd piłki

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Mamy n piłek baseballowych i n maszyn do ich obsługi. Każda piłka ma swoją wagę w_i, a każda maszyna ma swoją moc obsługi p_i. Możemy wybrać piłkę i maszynę do jej obsługi. Powiedzmy, że *trudność* złapania i-tej piłki obsługiwanej za pomocą maszyny j wynosi $w_i \times p_j$. Chcemy wybrać parę $(i; j)$ taką, że każda inna para $(i'; j')$ $(i \neq i'$ i $j \neq j')$ ma trudność złapania mniejszą lub równą trudności złapania pary $(i; j)$. Podaj liczbę takich par.

Input

Pierwsza linia wejścia zawiera pojedynczą liczbę całkowitą n $(1 \leq n \leq 2 \cdot 10^5)$ — liczbę piłek i maszyn do obsługi.

Druga linia wejścia zawiera n liczb całkowitych w_i $(1 \leq w_i \leq 10^9)$ — wagi piłek.

Trzecia linia wejścia zawiera n liczb całkowitych p_i $(1 \leq p_i \leq 10^9)$ — moce maszyn do obsługi.

Output

Wypisz jedną liczbę całkowitą — odpowiedź na problem.

Scoring

1. (7 punktów): $n \leq 3$;
2. (7 punktów): wszystkie w_i są równe i wszystkie p_i są równe;
3. (8 punktów): wszystkie w_i są równe;
4. (9 punktów): $w_i, p_i \leq 1000$;
5. (23 punkty): $n \leq 100$
6. (24 punkty): $n \leq 1000$
7. (22 punktów): brak dodatkowych ograniczeń.

Examples

standard input	standard output
2 1 3 2 4	2
4 1 5 9 6 7 4 5 3	2

Problem B. Klasyczny problem zapytań

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Dostajesz dwie liczby całkowite C i L oraz tablicę a składającą się z n liczb całkowitych. Masz q zapytań w formacie $l\ r\ x$. Po każdym zapytaniu musisz:

1. odjąć x od wszystkich elementów na przedziale $[l; r]$ tablicy a;
2. powiedzieć, czy istnieje `dobry` przedział S na tablicy a. Przedział S jest uważany za `dobry`, jeśli $len(S) \geq L$ i $avg(S) \leq C$, gdzie $len(S)$ oznacza długość przedziału S, a $avg(S)$ oznacza średnią wartość na przedziale S.

Input

Pierwsza linia wejścia zawiera cztery liczby całkowite n, L $(1 \leq L \leq n \leq 2 \cdot 10^5)$, q $(1 \leq q \leq 2 \cdot 10^5)$, C $(-10^9 \leq C \leq 10^9)$.

Druga linia wejścia zawiera n liczb całkowitych a_i $(-10^9 \leq a_i \leq 10^9)$ — dana tablica a.

Następne q linii zawierają po trzy liczby całkowite l, r $(1 \leq l \leq r \leq n)$, x $(0 \leq x \leq 10^7)$ każda — opis zapytań.

Output

Wypisz q linii, z których każda składa się z `YES`, jeśli istnieje `dobry` przedział S, lub `NO` w przeciwnym razie.

Scoring

1. (11 punktów): $n \leq 3$;
2. (12 punktów): $n \leq 100$, $q \leq 1000$;
3. (14 punktów): $q = 1$, $C = 0$;
4. (15 punktów): $C = 0$;
5. (19 punktów): $q = 1$;
6. (29 punktów): brak dodatkowych ograniczeń.

Example

standard input	standard output
4 2 3 2 5 6 3 4 1 2 2 1 3 1 1 4 2	NO NO YES

Problem C. Zapytania o drzewo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Dostajesz połączony graf z n wierzchołkami i $n-1$ krawędziami – innymi słowy, jest to drzewo. Wierzchołek 1 jest korzeniem drzewa. Na wierzchołku v jest zapisana liczba całkowita a_v. Koszt drzewa definiujemy jako operacja bitowa `XOR` wszystkich jego wartości. Dostajesz q zapytań, zawierających dwie liczby całkowite v i x. Każde zapytanie wykonuje $a_u = a_u \oplus x$ dla wszystkich u należących do poddrzewa wierzchołka v, a $\oplus$ oznacza operację `XOR`. Jak wiadomo, ten problem może być łatwo rozwiązany przez naszych uczestników, ale gdyby wszystko było takie łatwe, ten problem nie pojawiłby się na tej olimpiadzie. Wiesz, że niektóre zapytania zostały zmienione przez złośliwego Antona. Zatem prosisz o znalezienie sumy kosztów modulo 998244353, wszystkich 2^q drzew, które można uzyskać przez zastosowanie niektórych (być może żadnych) zapytań na drzewie.

Input

Pierwsza linia wejścia zawiera dwie liczby całkowite n ($1 \leq n \leq 2 \cdot 10^5$) i q ($1 \leq q \leq 2 \cdot 10^5$) — liczba wierzchołków w drzewie i liczba zapytań odpowiednio.

Druga linia wejścia zawiera n liczb całkowitych a_i ($0 \leq a_i < 2^{60}$) — wartości zapisane na wierzchołkach drzewa.

Następne $n-1$ linii zawierają po dwie liczby całkowite u i v każda — krawędzie danego drzewa.

Następne q linii zawierają po dwie liczby całkowite v ($1 \leq v \leq n$) i x ($0 \leq x < 2^{60}$) każda — opis zapytań.

Output

Wypisz jedną liczbę całkowitą w pierwszej linii — odpowiedź na problem.

Scoring

1. (7 punktów): $q \leq 2$;
2. (12 punktów): $q \leq 10$;
3. (16 punktów): $q \leq 20$;
4. (18 punktów): $a_i, x < 2$;
5. (13 punktów): $a_i, x < 1024$;
6. (15 punktów): $a_i, x < 2^{18}$;
7. (19 punktów): brak dodatkowych ograniczeń.

Example

standard input	standard output
5 3 1 0 2 3 1 1 2 1 3 3 4 3 5 3 2 4 3 1 5	28

Problem D. Anton - strażnik

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Anton jest strażnikiem odpowiedzialnym za n obiektów o numerach od 1 do n. Istnieje również $n-1$ dróg, gdzie i-ta droga łączy obiekty u_i i v_i i ma długość w_i. Z każdego obiektu można dotrzeć do obiektu 1.

Na początku Anton znajduje się w obiekcie 1.

Zdefiniujmy priorytet każdego obiektu jako minimalną liczbę dróg, które trzeba przebyć z tego obiektu do obiektu 1. Na przykład dla obiektu 1 ta liczba wynosi 0; dla wszystkich obiektów bezpośrednio połączonych z 1, wynosi 1, i tak dalej.

Anton musi odwiedzić wszystkie obiekty. Musi najpierw odwiedzić wszystkie obiekty, które mają priorytet 1, następnie te, które mają priorytet 2, i tak dalej. Jeśli istnieją obiekty o tym samym priorytecie, Anton może wybrać kolejność ich odwiedzania. Zauważ, że może odwiedzić każdy obiekt o priorytecie k tylko po odwiedzeniu wszystkich obiektów o priorytecie $k-1$.

Znajdź minimalną odległość, jaką musi przebyć.

Input

W pierwszej linii znajduje się liczba całkowita n ($1 \le n \le 10^6$).

W drugiej linii znajduje się $n-1$ liczb całkowitych p_i, które wskazują, że istnieje droga z obiektu $i+1$ do obiektu p_i ($1 \le p_i \le n$).

W trzeciej linii znajduje się $n-1$ liczb całkowitych w_i, które są długościami dróg między obiektami $i+1$ i p_i ($1 \le w_i \le 10^9$).

Zagwarantowane jest, że można dotrzeć do wszystkich innych obiektów z obiektu 1.

Output

Wyjście powinno zawierać pojedynczą liczbę całkowitą — odpowiedź na problem.

Scoring

1. (4 punkty): $1 \le n \le 10$;
2. (5 punktów): $1 \le n \le 22$;
3. (13 punktów): nie więcej niż 6 obiektów ma ten sam priorytet;
4. (10 punktów): nie więcej niż 10 obiektów ma ten sam priorytet;
5. (15 punktów): $w_i = 1$;
6. (15 punktów): $1 \le n \le 10^5$;
7. (38 punktów): bez dodatkowych ograniczeń.

Examples

standard input	standard output
7 3 1 1 1 5 6 14 10 6 5 7 3	85
5 1 2 3 4 5 6 10 21	42
4 1 1 1 4 10 30	58