Superalbero

Ti viene dato un albero radicato di n nodi, identificati dagli indici 0, ..., $n-1$. La radice ha indice 0. Per ogni $i \in \{0, \dots, n-1\}$, il nodo i (ovvero il nodo di indice i) ha assegnato un intero a_i. Sia f_v il valore dell'AND bitwise (denotato da $\&$) dei valori a_i nel percorso semplice dal nodo v alla radice. (Nota che il percorso semplice dal nodo x al nodo y include sia x che y). Sia la *potenza* dell'albero il valore di

$$\sum_{0 \le u,v < n} f_u \cdot f_v,$$

e sia la *superpotenza* dell'albero il valore di (nota la differenza negli indici)

$$\sum_{0 \le u < v < n} f_u \cdot f_v.$$

Per un esempio pratico, guarda la spiegazione dei casi di esempio sotto.

Diciamo che un nodo u appartiene al *sottoalbero di un nodo* v se v appartiene al percorso semplice dal nodo u alla radice. Nota che il sottoalbero di un nodo x include il nodo x stesso.

Ti vengono date q modifiche. Ogni modifica è descritta da due interi, v e x, e ti richiede di impostare $a_u := a_u \,\&\, x$ per ogni nodo u nel sottoalbero del nodo v. Dopo ogni modifica, devi stampare la potenza e la superpotenza dell'albero corrente.

Dato che la risposta può essere molto grande, stampala modulo $10^9 + 7$.

Formato di input

La prima riga in input contiene gli interi n e q.

La seconda riga in input contiene $n-1$ interi, p_1, p_2, ..., p_{n-1}, che determinano la struttura dell'albero. Per ogni $i \in \{1, \dots, n-1\}$, p_i è l'indice del padre del nodo i, e vale $0 \le p_i < i$.

La terza riga in input contiene n interi, a_0, a_1, ..., a_{n-1}. Questi sono i valori assegnati ai nodi.

Ognuna delle successive q righe contiene due interi, v ($0 \le v < n$) e x. Questi interi specificano le modifiche.

Formato di output

Stampa $q+1$ righe. Ogni riga deve contenere due interi separati da uno spazio. Nella prima riga stampa la potenza e la superpotenza (entrambe modulo 10^9+7) dell'albero iniziale. Nell'i-esima riga delle successive q righe ($i \in \{1, \ldots, q\}$), stampa la potenza e la superpotenza (entrambe modulo 10^9+7) dell'albero dopo la i-esima modifica.

Assunzioni

- $1 \leq n, q \leq 10^6$.
- $0 \leq a_i < 2^{60}$ per ogni $i \in \{0, \ldots, n-1\}$.
- $0 \leq x < 2^{60}$ per ogni modifica (v, x).

Punteggio

Per un caso di test, la tua soluzione riceverà 50% del punteggio se produce valori di potenza tutti corretti ma produce almeno un valore di superpotenza incorretto per quel caso di test.

Allo stesso modo, verrà assegnato 50% del punteggio di un caso di test a una soluzione che produce valori di superpotenza tutti corretti ma produce almeno un valore di potenza incorretto per quel caso di test.

Subtask

1. (4 punti) $n = 3$.
2. (7 punti) $n, q \leq 700$.
3. (13 punti) $n, q \leq 5000$.
4. (6 punti) $n \leq 10^5$, $p_i = i - 1$ (per ogni $i \in \{1, \ldots, n-1\}$), e $a_i, x < 2^{20}$ (per ogni $i \in \{0, \ldots, n-1\}$ e per ogni modifica (v, x)).
5. (7 punti) $p_i = i - 1$ (per ogni $i \in \{1, \ldots, n-1\}$).
6. (12 punti) $a_i, x < 2^{20}$ (per ogni $i \in \{0, \ldots, n-1\}$ e per ogni modifica (v, x)).
7. (14 punti) $n \leq 10^5$.
8. (11 punti) $n \leq 5 \cdot 10^5$.
9. (26 punti) Nessuna limitazione aggiuntiva.

Caso d'esempio 1

Input

```
3 3
0 0
7 3 4
1 6
2 2
0 3
```

Output

```
196 61
169 50
81 14
25 6
```

Spiegazione

Inizialmente abbiamo

$$f_0 = 7,\ f_1 = 7\&3 = 3,\ f_2 = 7\&4 = 4.$$

Quindi la potenza dell'albero è uguale a

$$f_0 \cdot f_0 + f_0 \cdot f_1 + f_0 \cdot f_2 + f_1 \cdot f_0 + f_1 \cdot f_1 + f_1 \cdot f_2 + f_2 \cdot f_0 + f_2 \cdot f_1 + f_2 \cdot f_2 =$$

$$= 7 \cdot 7 + 7 \cdot 3 + 7 \cdot 4 + 3 \cdot 7 + 3 \cdot 3 + 3 \cdot 4 + 4 \cdot 7 + 4 \cdot 3 + 4 \cdot 4 = 196.$$

La superpotenza è uguale a

$$f_0 \cdot f_1 + f_0 \cdot f_2 + f_1 \cdot f_2 = 7 \cdot 3 + 7 \cdot 4 + 3 \cdot 4 = 61.$$

Dopo la prima modifica:

$$a_0 = 7,\ a_1 = 3\&6 = 2,\ a_2 = 4;$$

$$f_0 = 7,\ f_1 = 2,\ f_2 = 4.$$

Dopo la seconda modifica:

$$a_0 = 7,\ a_1 = 2,\ a_2 = 4\&2 = 0;$$

$$f_0 = 7,\ f_1 = 2,\ f_2 = 0.$$

Dopo la terza modifica:

$$a_0 = 7\&3 = 3,\ a_1 = 2\&3 = 2,\ a_2 = 0\&3 = 0;$$

$$f_0 = 3,\ f_1 = 2,\ f_2 = 0.$$

Caso d'esempio 2

Input

```
4 2
0 0 1
6 5 6 2
1 2
0 3
```

Output

```
256 84
144 36
16 4
```

Spiegazione

Inizialmente abbiamo

$$f_0 = 6,\ f_1 = 6\&5 = 4,\ f_2 = 6\&6 = 6,\ f_3 = 2\&5\&6 = 0.$$

Dopo la prima modifica:

$$a_0 = 6,\ a_1 = 5\&2 = 0,\ a_2 = 6,\ a_3 = 2\&2 = 2;$$

$$f_0 = 6,\ f_1 = 0,\ f_2 = 6,\ f_3 = 2\&0 = 0.$$

Dopo la seconda modifica:

$$a_0 = 7,\ a_1 = 2,\ a_2 = 4\&2 = 0;$$

$$f_0 = 7,\ f_1 = 2,\ f_2 = 0.$$

Caso d'esempio 3

Input

```
7 3
0 0 1 1 2 2
7 6 5 7 3 4 2
4 4
3 3
2 1
```

Output

```
900 367
784 311
576 223
256 83
```