# Super Baum

Gegeben ist ein gewurzelter Baum mit $n$ Knoten, mit Indizes $0$, …, $n-1$. Die Wurzel hat den Index 0. Für $i \in \{0, \ldots, n-1\}$ wird dem Knoten $i$ (d.h. dem Knoten mit dem Index $i$) eine ganze Zahl $a_i$ zugewiesen. Sei $f_v$ der Wert des bitweisen AND (fortan mit $\&$ bezeichnet) der Werte $a_i$ auf dem Pfad vom Knoten $v$ zur Wurzel. (Beachte, dass der Pfad von einem Knoten $x$ zu einem Knoten $y$ sowohl $x$ als auch $y$ einschliesst.)

Die *Stärke* des Baums ist definiert als

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

und die *Superstärke* des Baums ist definiert als

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

(beachte den Unterschied in den Bereichen von $u$ und $v$).

Für ein erläuterndes Beispiel siehe die Erklärung der Beispieltestfälle unten.

Wir sagen, dass ein Knoten $u$ zum *Teilbaum eines Knotens* $v$ gehört, wenn $v$ zum Pfad vom Knoten $u$ zur Wurzel gehört. Beachte, dass der Teilbaum eines Knotens $x$ den Knoten $x$ selbst einschliesst.

Es werden $q$ Updates vorgelegt. Jedes Update wird durch zwei ganze Zahlen, $v$ und $x$, beschrieben und verlangt $a_u := a_u \,\&\, x$ für jeden Knoten $u$ im Teilbaum des Knotens $v$ zu setzen. Nach jedem Update soll die Stärke und Superstärke des aktuellen Baums ausgeben werden.

Da die Ausgabewerte gross sein können, gib sie modulo $10^9 + 7$ aus.

## Eingabeformat

Die erste Zeile der Eingabe enthält die ganzen Zahlen $n$ und $q$.

Die zweite Zeile der Eingabe enthält $n-1$ Ganzzahlen, nämlich $p_1$, $p_2$, …, $p_{n-1}$, welche die Struktur des Baums bestimmen. Für $i \in \{1, \ldots, n-1\}$ ist $p_i$ der Index des Elternknotens von Knoten $i$, und es gilt $0 \leq p_i < i$.

Die dritte Zeile der Eingabe enthält $n$ ganze Zahlen, nämlich $a_0$, $a_1$, …, $a_{n-1}$. Dies sind die den Knoten zugewiesenen Werte.

Jede der folgenden $q$ Zeilen enthält zwei ganze Zahlen, $v$ ($0 \leq v < n$) und $x$. Diese beschreiben die einzelnen Updates.

## Ausgabeformat

Gib $q + 1$ Zeilen aus. Jede Zeile sollte zwei, durch ein Leerzeichen getrennte, ganze Zahlen enthalten. In der ersten Zeile gib die Stärke und die Superstärke (modulo $10^9 + 7$) des ursprünglichen Baums aus. In der $i$-ten Zeile, der verbleibenden $q$ Zeilen ($i \in \{1, \ldots, q\}$), gib die Stärke und die Superstärke (modulo $10^9 + 7$) des Baums nach dem $i$-ten Update aus.

## Limits

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

## Bewertung

Für einen gegebenen Testfall erhältst du $50\%$ der Punkte, wenn du alle Stärkewerte korrekt berechnest, aber mindestens einen Superstärkewert für diesen Testfall falsch berechnest.

Ebenso werden $50\%$ der Punkte für einen gegebenen Testfall vergeben, wenn du alle Superstärkewerte für diesen Testfall korrekt berechnest, aber mindestens einen Stärkewert falsch berechnest.

## Teilaufgaben

1. (4 Punkte) $n = 3$.
2. (7 Punkte) $n, q \leq 700$.
3. (13 Punkte) $n, q \leq 5000$.
4. (6 Punkte) $n \leq 10^5$, $p_i = i - 1$ (für $i \in \{1, \ldots, n-1\}$), und $a_i, x < 2^{20}$ (für $i \in \{0, \ldots, n-1\}$ und für jedes Update $(v, x)$).
5. (7 Punkte) $p_i = i - 1$ (für $i \in \{1, \ldots, n-1\}$).
6. (12 Punkte) $a_i, x < 2^{20}$ (für $i \in \{0, \ldots, n-1\}$ und für jedes Update $(v, x)$).
7. (14 Punkte) $n \leq 10^5$.
8. (11 Punkte) $n \leq 5 \cdot 10^5$.
9. (26 Punkte) Keine zusätzlichen Einschränkungen.

# Beispieltestfall 1

## Eingabe

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

## Ausgabe

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

## Erklärung

Zu Beginn haben wir

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

Deshalb ist die Stärke des Baums gleich

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

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

Die Superstärke entspricht

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

Nach dem ersten Update:

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

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

Nach dem zweiten Update:

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

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

Nach dem dritten Update:

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

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

## Testfall Beispiel 2

### Eingabe

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

### Ausgabe

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

### Erklärung

Zu Beginn haben wir

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

Nach dem ersten Update:

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

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

Nach dem zweiten Update:

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

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

# Testfall Beispiel 3

## Eingabe

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

## Ausgabe

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