

# Zadanie: PAR

## Park Bitowy

Laboratorium z ASD, lab 10. Dostępna pamięć: 240 MB.

12.01.2025, 23:59:59

W Parku Bitowym znajduje się  $n$  polanek ponumerowanych od 1 do  $n$ . Niektóre pary polanek są połączone (dwukierunkowymi) ścieżkami. Jak to przystało na park bitowy, układ ścieżek tworzy drzewo binarne, którego korzeniem jest polanka numer 1.

Bajtek i Bajtyna przyszli po szkole pobawić się w parku. Dzieci postanowiły zagrać w następującą grę. Naprzemiennie jedno z dzieci wskazuje numer polanki  $a$  oraz liczbą całkowitą nieujemną  $d$ , a zadaniem drugiego z nich jest odnalezienie w parku jakiejś polanki, której odległość od polanki  $a$  wynosi  $d$ . Jeśli takiej polanki nie ma, dziecko musi to określić.

Bajtek chciałby sobie ułatwić grę. Poprosił Cię, żebyś napisał program, który będzie odnajdował polanki określone przez Bajtynę.

### Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita  $n$  ( $2 \leq n \leq 500\,000$ ), oznaczająca liczbę polanek w Parku Bitowym. Każdy z kolejnych  $n$  wierszy zawiera dwie liczby całkowite  $a_i$  oraz  $b_i$  ( $a_i, b_i \in \{-1, 1, 2, \dots, n\}$ ), oznaczające, że z polanki numer  $i$  prowadzą ścieżki na polanki numer  $a_i$  oraz  $b_i$ . Wartość  $-1$  oznacza, że dana ścieżka nie istnieje. Dane wejściowe zawierają wszystkie krawędzie konieczne do jednoznacznego zbudowania drzewa ukorzonego w polance numer 1.

W kolejnym wierszu wejścia znajduje się jedna liczba całkowita  $m$  ( $1 \leq m \leq 500\,000$ ), oznaczająca liczbę poleceń, które Bajtek otrzymał od Bajtyny. Każdy z następnych  $m$  wierszy zawiera dwie liczby całkowite  $a$  oraz  $d$  ( $1 \leq a \leq n$ ,  $0 \leq d < n$ ).

### Wyjście

Twój program powinien wypisać numery polanek stanowiące odpowiedzi na pytania Bajtyny. Jeśli odpowiedzią na dane pytanie jest więcej niż jedna polanka, Twój program powinien wypisać jakąkolwiek z nich. Jeśli polanka wskazana przez Bajtynę nie istnieje, w odpowiednim wierszu należy wypisać  $-1$ .

### Przykład

Dla danych wejściowych:

```
8
3 4
-1 6
2 5
-1 -1
7 -1
-1 -1
8 -1
-1 -1
6
1 3
1 4
1 5
6 1
6 4
6 5
```

poprawnym wynikiem jest:

```
6
8
-1
2
4
8
```

## Wskazówki

- Pokaż, jak dla każdego wierzchołka  $u$  i dla każdego całkowitego  $0 \leq i \leq \log n$  wyznaczyć wierzchołek położony  $2^i$  krawędzi ponad wierzchołkiem  $u$ .
- Sprowadź całe zadanie do następującego: dla danych dwóch wierzchołków  $u$  i  $v$ , znajdź wierzchołek położony na ścieżce między tymi wierzchołkami, którego odległość od  $u$  wynosi  $d_u$ , a odległość od  $v$  wynosi  $d_v$ . W sprowadzeniu może Ci być potrzebne proste programowanie dynamiczne na drzewie.
- Rozwiązanie wzorcowe ma złożoność czasową i pamięciową  $O(n \log n)$ .

## Rysunek drzewa z przykładu

