C. Fun Tour

Time limit	2 s
Memory limit	512 MB

Description

Жакартанын эң чоң паркында N аттракцион бар жана алар 0 дөн $N-1$ге чейин номерленген. Бул аттракциондор $N-1$ эки тараптуу жолдор менен туташтырылган, ошондуктан каалаган эки аттракциондун арасында уникалдуу жол бар. Жолдор 0 дөн $N-2$ чейин номерленген. i - жол $A[i]$ - инчи жана $B[i]$ -инчи аттракциондорду байланыштырат жана бул жолду басып өтүү үчүн бир саат талап кылынат. Тыгындарды болтурбоо үчүн, ар бир аттракцион эң көп дегенде үч жолдун чекити болуп саналат.

Бардык аттракциондорду бир жолуда гана саякат жасагыңыз келет.Бир жолдон эки же андан көпжолу өтүү кызыксыз. Көңүлдүү саякат жасоо үчүн, бардык аттракциондордун иреттүүлүгүн тапкыңыз келет, анткени кийинки аттракционго баруу үчүн талап кылынган убакыт мурунку аттракционго барган убакыттан ашпашы керек. Башкача айтканда, P 0, P [1], \ ldots, P [N-1] 0 ден чейин бүтүн сандарды камтыган бир катарды тапкыңыз келет. N - 1 биржолу, P [i] −читартуудан P [i + 1] −тартуугакеткенубакытузакэмесболгондой P [i-1] −читартуудан P [i] −тартууга, 0 \ lt i \ lt N үчүн - талап кылынган убакытка караганда, 1 $.

Сизде кызыктуу жерлердин толук картасы жок. Андыктан, кызыктуу экскурсия түзүү үчүн маалымат борборуна бир нече суроолорду беришиңиз керек. $0\ leX, Y\ ltN$ бар болгону эки параметрден турган Q жолу суроолорду берсеңиз болот. Ар бир суроо төмөнкүлөрдүн бири:

- X - инчи аттракциондон Y - инчи аттракционго чейин канча саат талап кылынат? Атап айтканда, $X=Y$ болсо, анда 0 жооп берилет.
- Канча түрдүү Z бар X - инчиаттракциондон Z кебаруууүчүн Y −аркылууөтөт.Тактапайтканда, X = Y болсо, анда N $ жооп берилет.

Task

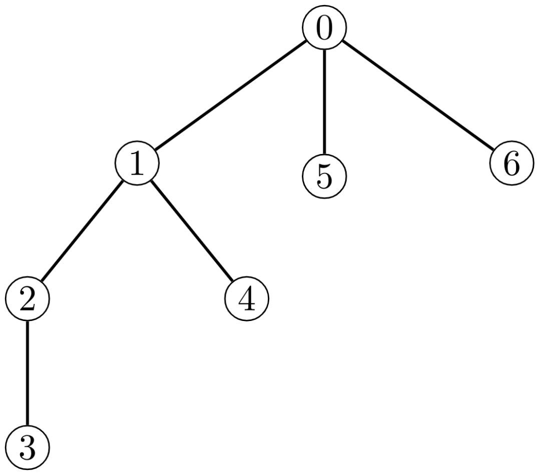
You have to implement `createFunTour` function:

- `createFunTour(N, Q)` - This function will be called by the grader exactly once.
  - N: An integer representing the number of attractions.
  - Q: An integer representing the maximum number of questions.
  - This function is allowed to call two grader functions:
    - `hoursRequired(X, Y)`
      - X: An integer representing the first attraction.
      - Y: An integer representing the second attraction.
      - This functions returns an integer representing hours required to go from the X-th attraction to the Y-th attraction.
      - If either X or Y is not an integer between 0 and $N-1$, then you will get a WA verdict.
    - `attractionsBehind(X, Y)`
      - X: An integer representing the first attraction.
      - Y: An integer representing the second attraction.
      - This function returns an integer representing the number of attractions Z such that you have to visit the Y-th attraction to go from the X-th attraction to the Z-th attraction.
      - If either X or Y is not an integer between 0 and $N-1$, then you will get a WA verdict.

- This function must return an array of N integers representing the permutation of attractions in a fun tour.

Example

In the following example, $N = 7$, $Q = 400\,000$, $A = [0, 0, 0, 1, 1, 2]$, and $B = [1, 5, 6, 2, 4, 3]$. The example is illustrated by the following image:



Grader will call `createFunTour(7, 400000)`.

- If the contestant queries `hoursRequired(3, 5)`, then the function will return 4.
- If the contestant queries `hoursRequired(5, 4)`, then the function will return 3.
- If the contestant queries `attractionsBehind(5, 1)`, then the function will return 4. To go from the fifth attraction to the first, second, third, and fourth attractions, you will have to visit the first attraction.
- If the contestant queries `attractionsBehind(1, 5)`, then the function will return 1.
- The contestant can return $[3, 6, 4, 5, 2, 0, 1]$ since the hours required to visit the next attractions are $[4, 3, 3, 3, 2, 1]$ in order.

Constraints

- $2 \le N \le 100\,000$.
- $Q = 400\,000$.
- It is possible to travel between any pair of attractions through the roads.
- Each attraction is an endpoint of at most three roads.

Subtask 1 (10 points)

- $N \le 17$.

Subtask 2 (16 points)

- $N \le 500$.

Subtask 3 (21 points)

- There is a road connecting the i-th attraction and the $\lfloor \frac{i-1}{2} \rfloor$-th attraction, for all $1 \le i < N$.

Subtask 4 (19 points)

- There is at least an attraction T such that for all $0 \le i < N$, `hoursRequired(T, i)` < 30 and there exists an interval $[L[i], R[i]]$ $(0 \le L[i] \le i \le R[i] < N)$ satisfying the following conditions:

- You have to visit the i-th attraction to go from the T-th attraction to the j-th attraction if and only if $L[i] \le j \le R[i]$.
- If $L[i] < i$, then there must be exactly one attraction X such that:
  - $L[i] \le X < i$.
  - There is a road connecting the i-th attraction and the X-th attraction.
- If $i < R[i]$, then there must be exactly one attraction Y such that:
  - $i < Y \le R[i]$.
  - There is a road connecting the i-th attraction and the Y-th attraction.

Subtask 5 (34 points)

- No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

```
N Q
A[0] B[0]
A[1] B[1]
.
.
.
A[N-2] B[N-2]
```

The sample grader writes the integers returned by `createFunTour` if it correctly returns an array of N integers representing the permutation of attractions in a fun tour and calls both `hoursRequired` and `attractionsBehind` not more than Q times combined. Otherwise, it prints a wrong answer message.