

C. 楽しいツアー (Fun Tour)

Time limit	2 s
Memory limit	512 MB

問題文

ジャカルタ内最大のテーマパークには、0 から $N - 1$ までの番号が付けられた、 N 個のアトラクションが存在する。これらのアトラクションは $N - 1$ 本の双方向に通行可能な道路によって繋がれており、任意の 2 つのアトラクションのペアについて、一方のアトラクションから出発して、同じアトラクションを 2 回以上通らずに、道路をいくつか経由してもう一方のアトラクションに到着する方法が、ちょうど 1 通り存在する。道路には、0 から $N - 2$ までの番号が付けられている。道路 i は、アトラクション $A[i]$ とアトラクション $B[i]$ を繋いでいて、端から端まで歩くには 1 時間かかる。混雑を避けるため、各アトラクションは 3 本以下の道路としか繋がっていない。

あなたは、すべてのアトラクションをちょうど 1 回ずつ訪れるようなツアーを作ろうとしている。あるアトラクションから次のアトラクションに移動するために、たくさんの道路を通るのは退屈である。あなたは、「楽しいツアー」を作るために、次の条件を満たすようにアトラクションを訪れる順番を工夫したい: 現在のアトラクションから次のアトラクションへの移動にかかる時間が、1 つ前のアトラクションから現在のアトラクションへの移動にかかる時間以下になるようにする。すなわち、あなたはある配列 $P[0], P[1], \dots, P[N - 1]$ を見つけたい。この配列は、0 以上 $N - 1$ 以下の整数をちょうど 1 個ずつ含み、各 $0 < i < N - 1$ について、アトラクション $P[i]$ からアトラクション $P[i + 1]$ への移動にかかる時間は、アトラクション $P[i - 1]$ からアトラクション $P[i]$ への移動にかかる時間以下である。

あなたが所持しているアトラクションの地図は不完全である。そのため、「楽しいツアー」を作るために、あなたは案内所に何回か質問をする必要がある。あなたは Q 回以下の質問をすることができ、各質問では、2 つのパラメーター X と Y ($0 \leq X, Y < N$) を送ることができる。各質問は以下のどちらかである:

- アトラクション X からアトラクション Y への移動に何時間かかるか? 特に、 $X = Y$ のときは、答えは 0 である。
- アトラクション X からアトラクション Z に移動するとき、アトラクション Y を通る必要があるような、アトラクション Z は何個か? ただし、この個数には、アトラクション Y も含むことにする。特に、 $X = Y$ のとき、答えは N である。

課題

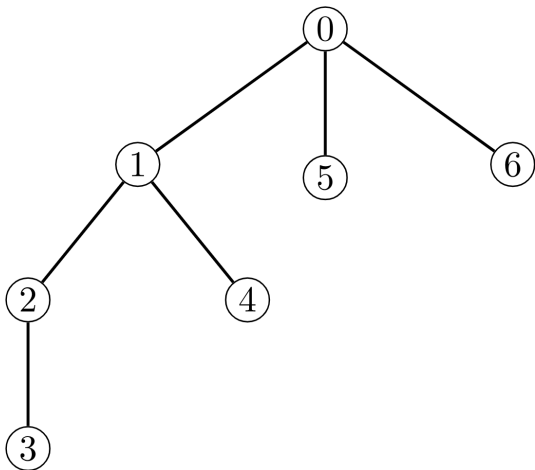
関数 `createFunTour` を実装せよ:

- `createFunTour(N, Q)` - この関数は、採点プログラム (grader) によってちょうど 1 回呼び出される。
 - N : アトラクションの個数を表す整数。
 - Q : 質問の上限回数を表す整数。
 - この関数は以下の 2 つの関数を呼ぶことができる:
 - `hoursRequired(X, Y)`
 - X : 1 つ目のアトラクションを表す整数。
 - Y : 2 つ目のアトラクションを表す整数。
 - この関数はアトラクション X からアトラクション Y への移動にかかる時間を表す整数を返す。
 - もし、 X または Y が 0 以上 $N - 1$ 以下の整数ではなかったとき、WA (不正解) と判定される。
 - `attractionsBehind(X, Y)`
 - X : 1 つ目のアトラクションを表す整数。
 - Y : 2 つ目のアトラクションを表す整数。

- この関数はアトラクション X からアトラクション Z に移動するときに、アトラクション Y を通る必要があるような、アトラクション Z の個数を表す整数を返す。
- もし、 X または Y が 0 以上 $N - 1$ 以下の整数ではなかったとき、WA (不正解) と判定される。
- この関数は N 個の整数からなる配列を返さなければならない。この配列は、「楽しいツアー」におけるアトラクションを訪れる順番を表している。

例

以下の例は、 $N = 7, Q = 400\,000, A = [0, 0, 0, 1, 1, 2], B = [1, 5, 6, 2, 4, 3]$ である。下の図は、この例を図示したものである：



採点プログラムは `createFunTour(7, 400000)` を呼び出す。

- もし、`hoursRequired(3, 5)` を呼び出した場合、戻り値は 4 である。
- もし、`hoursRequired(5, 4)` を呼び出した場合、戻り値は 3 である。
- もし、`attractionsBehind(5, 1)` を呼び出した場合、戻り値は 4 である。アトラクション 5 からアトラクション 1, 2, 3, 4 へ移動するためには、アトラクション 1 を通る必要がある。
- もし、`attractionsBehind(1, 5)` を呼び出した場合、戻り値は 1 である。
- この関数は `[3, 6, 4, 5, 2, 0, 1]` を戻り値として返すことができる。この場合、あるアトラクションから次のアトラクションへの移動にかかる時間は、順番に、`[4, 3, 3, 3, 2, 1]` なので問題文の条件を満たす。

制約

- $2 \leq N \leq 100\,000$.
- $Q = 400\,000$.
- 1 本以上の道路を通ることで、任意の 2 つのアトラクションのペアについて、互いに行き来することができる。
- 各アトラクションは、3 本以下の道路としか繋がっていない。

小課題 1 (10 点)

- $N \leq 17$.

小課題 2 (16 点)

- $N \leq 500$.

小課題 3 (21 点)

- 各 $1 \leq i < N$ について、アトラクション i とアトラクション $\lfloor \frac{i+1}{2} \rfloor$ を結ぶ道路が存在する (ただし、 $\lfloor x \rfloor$ は、 x 以下の最大の整数を表す).

小課題 4 (19 点)

- 次のような性質を持つアトラクション T が 1 つ以上存在する: すべての $0 \leq i < N$ について、`hoursRequired(T, i) < 30` が成り立ち、かつ、以下の条件を満たすような区間 $[L[i], R[i]]$ ($0 \leq L[i] \leq i \leq R[i] < N$) が存在する:
 - アトラクション T からアトラクション j へ移動するとき、アトラクション i を通るための必要十分条件は、 $L[i] \leq j \leq R[i]$ である.
 - $L[i] < i$ のとき、以下の条件を満たすようなアトラクション X が、ちょうど 1 つ存在する:
 - $L[i] \leq X < i$.
 - アトラクション i とアトラクション X を繋ぐ道路が存在する.
 - $i < R[i]$ のとき、以下の条件を満たすようなアトラクション Y が、ちょうど 1 つ存在する:
 - $i < Y \leq R[i]$.
 - アトラクション i とアトラクション Y を繋ぐ道路が存在する.

小課題 5 (34 点)

- 追加の制約はない.

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を受け取る:

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

もし `createFunTour` が、「楽しいツアー」におけるアトラクションを訪れる順番を N 個の整数からなる配列として正しく返すことができ、かつ、`hoursRequired` と `attractionsBehind` の呼び出し回数の合計が Q 回以下だったとき、採点プログラムのサンプルは `createFunTour` の戻り値を出力する. そうではないとき、`wrong answer` (不正解) のメッセージを出力する.