K: Odd trip plans

原案：riantkb
問題文：climpet
データセット：riantkb
解答：hos, kotatsugame, riantkb

問題概要

- N 頂点 M 辺の無向グラフが与えられる
- 以下の 2 種類のクエリが合計 Q 個与えられるので処理しろ
    - 1 u v
        - (u, v) が存在するかどうかを反転する
    - 2 u v
        - u から v への walk で、全ての頂点に奇数回訪れるものが存在するか判定

- N, M, Q <= 10^5

考察

- ひとまず判定クエリ 1 個を解くことを考える

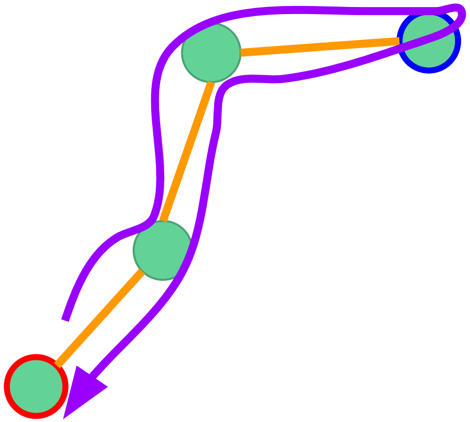
- 自明に、全体が連結でないと不可能
    - 以下は全体が連結であるとする

- 適当な walk を 1 つとった後、偶数回の頂点を奇数回に変えていけないか？

考察

- **今いる頂点**（適当にとった walk の終点）から**ある頂点**まで行って帰ってくると、**今いる頂点とある頂点**の偶奇だけ反転する

- 今いる頂点を S としたとき、
S → A → S → B → S と移動することで
A と B の 2 頂点のみの偶奇が反転する
    - A = S のときも成り立つ



- よって、少なくとも偶数回の頂点が偶数個であるときは条件が満たせる

考察

- 偶数回の頂点が偶数個である u から v の walk が作れますか？
- 以下では、「walk の長さ」を「頂点列の長さ」で定義します
  - 通った辺の本数とは偶奇が反転するので注意

- 補題：u から v の長さ k の walk について、
  「奇数回訪れた頂点の個数の偶奇」と「k の偶奇」は一致する
  - 各頂点の訪問回数の総和がk と一致するので、場合分けすれば証明できる

- よって、N の偶奇によって条件が変わる
  - N が偶数のとき：偶数長の長さのwalk が作れれば可能
  - N が奇数のとき：奇数長の長さのwalk が作れれば可能

考察

- 偶数または奇数の walk が作れるか？
- u から v の最短路の長さの偶奇がそもそも欲しいものであったら問題ない
- そうでなければ、偶奇が反転できる必要がある
  - これは、奇閉路があるかどうか（二部グラフかどうか）と同値
- 以上をまとめると
  - N が偶数のとき：奇閉路があるまたはu, v が二部グラフで異なる側のときに可能
  - N が奇数のとき：奇閉路があるまたはu, v の二部グラフで同じ側のときに可能

考察

- 残りは全て不可能か？

- 二部グラフなので、walk の偶奇を変えられない
- さっき使った「補題：u から v の長さ k の walk について、
「奇数回訪れた頂点の個数の偶奇」と「k の偶奇」は一致する」により、
奇数回訪れた頂点の個数の偶奇も変えられない

- つまり不可能であるとわかる

考察

- これでクエリ 1 つに答えることができた
  - 具体的には N の偶奇、二部グラフかどうか、二部グラフのときu, v が同じ側に属しているかどうかがわかればYesNo のどちらであるかがわかる

- 辺の削除を無視すると、例えば以下をするとできる
  - 頂点数 2N の union-find を用意する
  - 辺 (u, v) の追加については、(2u, 2v+1) と (2u+1, 2v) をそれぞれ union する
  - (u, v) の判定クエリについては以下を調べればよい
    - 連結成分数が 1, 2, 3以上 のどれか
    - (0, 1) が連結かどうか
      - 連結成分数が 2 だけど全体連結でない場合を判定するため
    - (2u, 2v) が連結かどうか

考察

- どうすれば辺が削除できる？

- 以下の 2 つの方針がある
  - 1. offline dynamic connectivity
  - 2. クエリ平方分割

  - 類題：Graph Construction（JAG 夏合宿 2010 : day3D）

700	Graph Construction (☆☆☆☆)	夏合宿 2010:day3D	92

考察

- offline dynamic connectivity
  - クエリ先読みができるときの 辺の追加・削除・連結判定クエリは O(log^2 N) とかで実現できる
  - offline dynamic connectivity でググると多分たくさんヒットします

- クエリ平方分割
  - B 個のクエリに対し、その中で変更のある辺は高々O(B) 本
    - 他の頂点を圧縮してしまえば、B 頂点 B 辺の追加削除になって、毎回愚直にやっても O(B α(B))
  - B = sqrt(Q) くらいにすると計算量はO(Q sqrt(Q) α(Q)) とか
    - 実際には B=1000 くらいにするのが速いです（バケットサイズガチャはしよう）

解法

- offline dynamic connectivity またはクエリ平方分割などで、
  辺の追加・削除・連結判定を高速に行う

- その上で、N の偶奇に注意しながら YesNo 判定をする
  - 連結成分数 2 だからといって全体連結かつ二部グラフとは限らないことに注意

統計情報

- Acceptances
  - 4 + 1 teams
- First Acceptance
  - Speed Star (72 min)