ICPC模擬国内予選2022
H: パレード

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

問題概要

- N 頂点 M 辺の単純無向連結グラフが与えられる。
- 次の条件を満たす walk が存在するならば、1 つ求めよ。
  - すべての辺を 1 回または 2 回だけ通る。
  - 同じ辺を 2 回連続で通らない。

制約

- $N, M \leq 20000$
  - 入力ファイルサイズを抑えるための制約で、本質的ではありません

次数 1 の頂点について

- 求める walk の始点と終点をそれぞれ S, G とする。
- 次数 1 の頂点に着目すると、明らかに次のことがわかる。
  - 次数 1 の頂点が 3 つ以上ある場合、解なし。
  - 次数 1 の頂点が 2 つある場合、片方が S でもう片方が G。
  - 次数 1 の頂点が 1 つだけの場合、S か G の少なくとも一方はその頂点。
  - 次数 1 の頂点がない場合、特に制約なし。
- 上記の制約を満たすよう、S と G を一対求める。
  - そのような頂点対は一般に複数ありえるが、どれでもよい。

各辺を通る回数の決め方

- 求めたい walk はオイラー路のようなものなので、2 回通る辺 (つまり倍化する辺) を決めて、オイラー路の存在条件を満たすようにする。具体的には
  - S と G が異なる頂点なら、S と G の次数は奇数。
  - 他の頂点の次数は偶数。
- 決め方は何でもよい。一例として、「S から G への経路を一つ取り、その経路上の辺は 1 回、残りの辺は 2 回通る」などとすればよい。

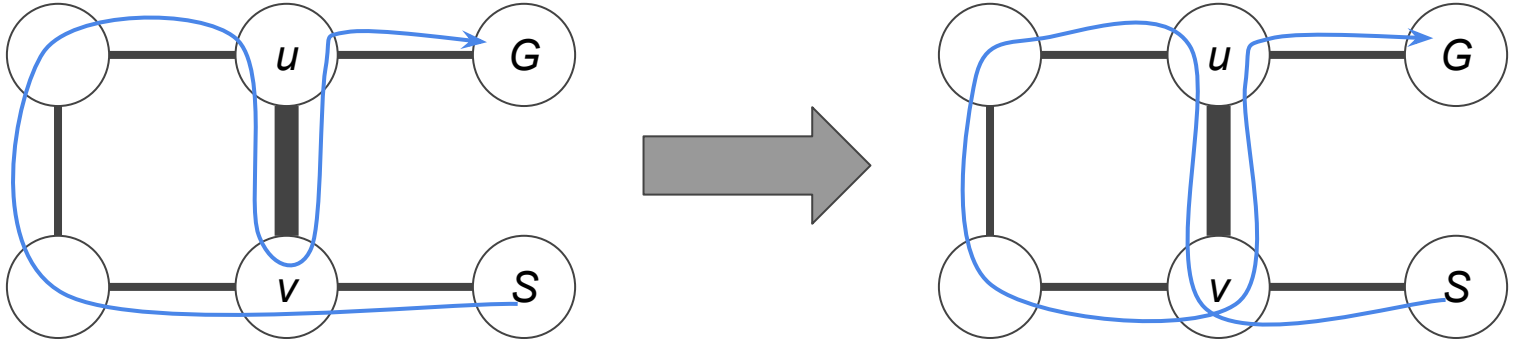
オイラー路の求め方

- 各辺を通る回数を決めた後、S から G へのオイラー路を 1 つ求める。
- 求め方は何でもよいが、実装の楽な方法として Hierholzer のアルゴリズムがある。以下の擬似コードについて dfs(G) を呼ぶと、G で終わるオイラー路の 1 つについて、頂点列が順に出力される。

```
dfs(u)
    for each v
        if 辺 (u, v) をまだ通れる
            dfs(v)
    output(u)
```

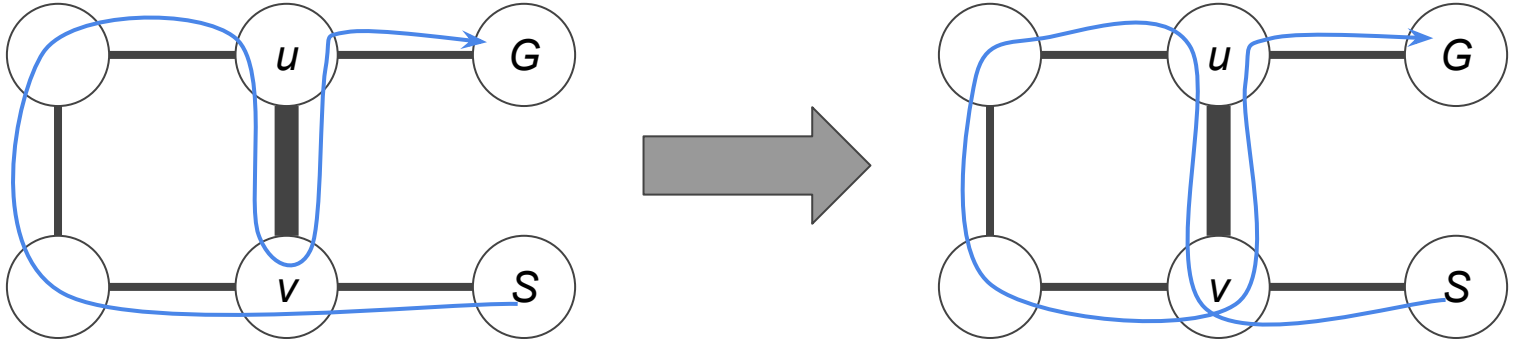
walk の修正

- 求めた walk には、同じ辺を 2 回連続で通る部分が含まれる場合がある。
  - このような部分を「折り返し」と呼ぶことにする。
- $u \to v \to u$ のような折り返しが存在する場合、この walk 上に頂点 v は少なくとも 2 回出現する。つまり、この walk 上に、v を含む有向閉路が存在する。
- その閉路の向きを反転させても、walk が崩れることはなく、かつ折り返しが一つ解消される。



walk の修正

- 折り返しがなくなるまで、閉路を見つけて反転させることを繰り返せばよい。
- 反転を愚直に行うと、計算量は最悪で $O(M^2)$ 時間となる。
  - とはいえ、実際に 2 乗時間かかる入力を作ることは難しいと思われる。



別解：折り返しのない walk を直接求める

- Hierholzer のアルゴリズムの動作を考えると、S から G への walk が折り返し $u \to v \to u$ を含むのは、DFS 中に次の状況に陥った場合である。
  - (u, v) は 2 回通ることにした辺であり、かつ、まだ1 回も通っていない。
  - 頂点 v を既に訪れたことがある。
  - v に接続する辺のうち、まだ通れる辺が(u, v) しか残っていない。
- 初めて v を訪れたとき、DFS で次に選ぶ辺を (u, v) にすることで、1 番目の条件を回避することができる。
- つまり、DFS 中に次に選ぶ辺を、次のルールに沿って決めれば、折り返しのない walk が直接求められる。
  - あと 2 回通る辺が存在するならば、そのような辺を優先的に選ぶ。
  - 直前に通った辺は選ばない。
- 全体で $O(M)$ 時間で実装できる。

ジャッジ解

- climpet (方針1): 116行, 1938 bytes
- climpet (方針2): 115行, 1921 bytes
- hos (方針2): 191行, 4301 bytes
- rian (方針1): 107行, 3015 bytes
- rian (方針2): 104行, 2891 bytes

統計情報

- AC / trying teams
  - 0 / 0
- First acceptance
  - -