

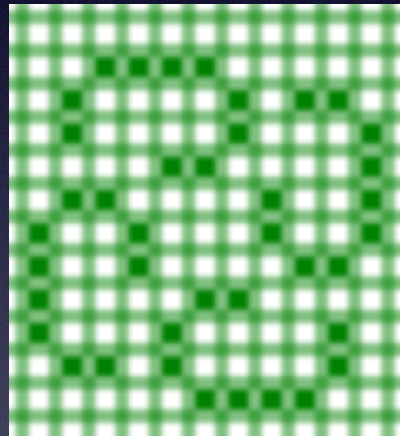
オイラー路

Rust

bit

北大数学科 B3 浅野

飽きたら
ライフゲームで遊んでみてください



<https://hei-7b848.firebaseio.com/bokulife/index.html>

おしながき

- ・ 自己紹介
- ・ オイラー路
- ・ Rustを使ってみませんか
- ・ おまけ: 明日から使えるbit abs min

おしながき

- ・ 自己紹介
- ・ オイラー路
- ・ Rustを使ってみませんか
- ・ おまけ: 明日から使えるbit abs min

自己紹介

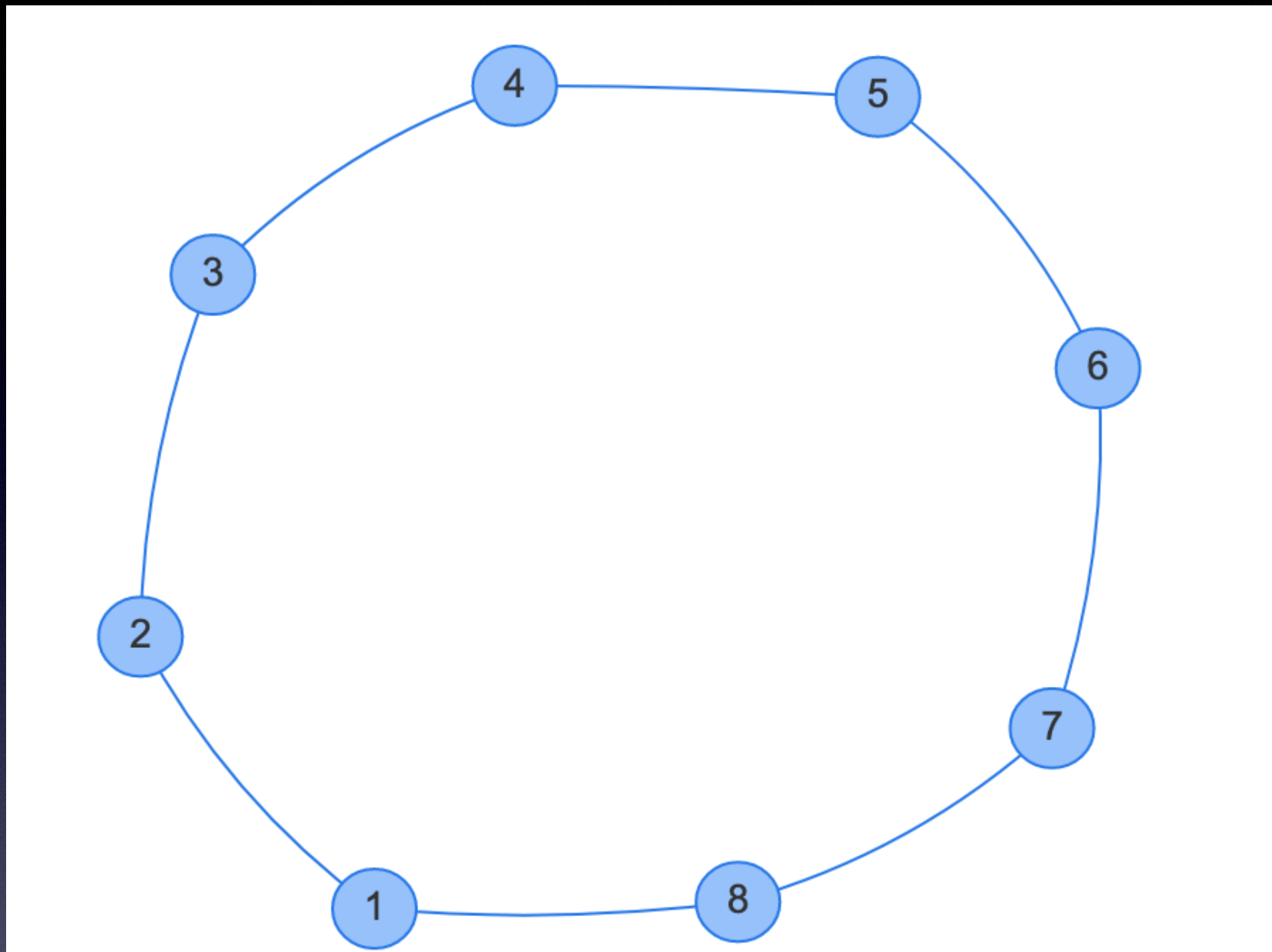
- ・ 北大数学科 B3
- ・ 前はauひかりを売っていた
- ・ 学部ではグレブナー基底の計算(代数幾何)を勉強しています
- ・ 最近の趣味
 - ・ 本を読む
 - ・ ご飯を作る
 - ・ 適当なコードを書く

おしながき

- ・ 自己紹介
- ・ **オイラー路**
- ・ Rustを使ってみませんか
- ・ おまけ: 明日から使えるbit abs min

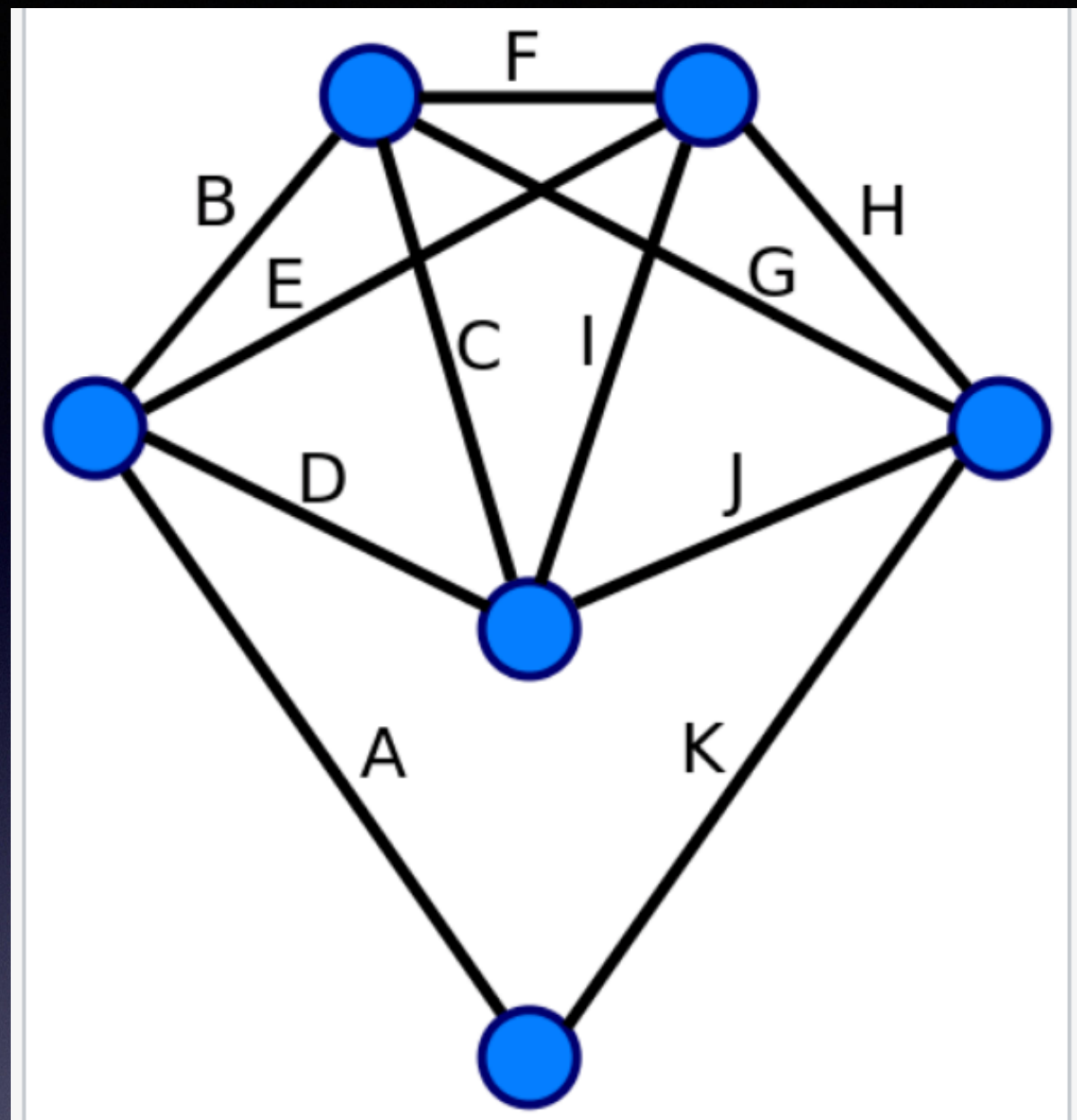
オイラー路の勉強

- ・ Wikipediaは下のlinkをどうぞ
 - ・ <https://ja.wikipedia.org/wiki/オイラー路>
- ・ 予備校のノリで学びたい人は下のlinkをどうぞ
 - ・ <https://www.youtube.com/watch?v=PFcuUpBwgtE>
- ・ 高校数学の美しい物語は下のlinkをどうぞ
 - ・ https://mathtrain.jp/euler_graph

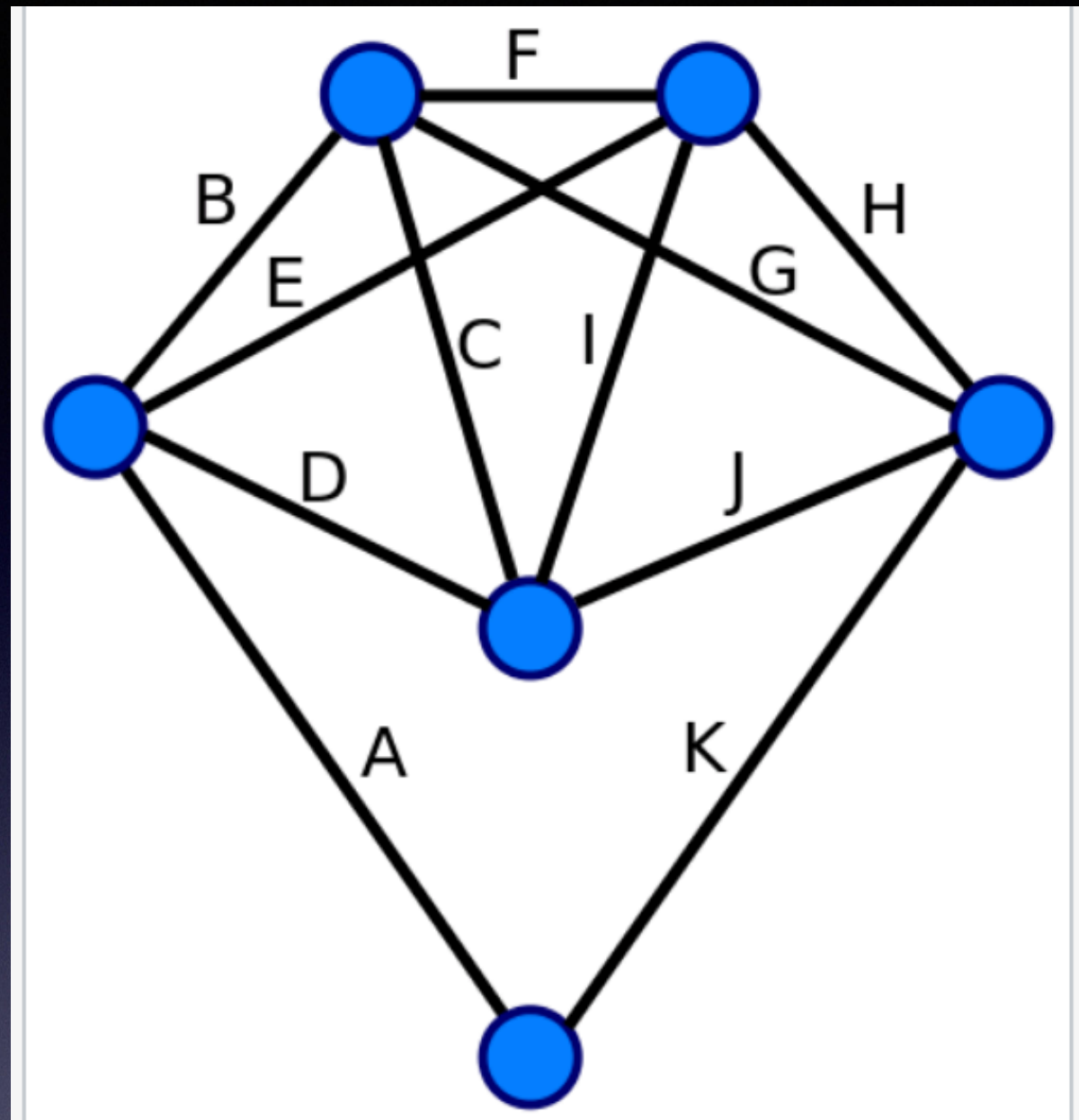


一筆書きできますか？

僕はできます(自明)

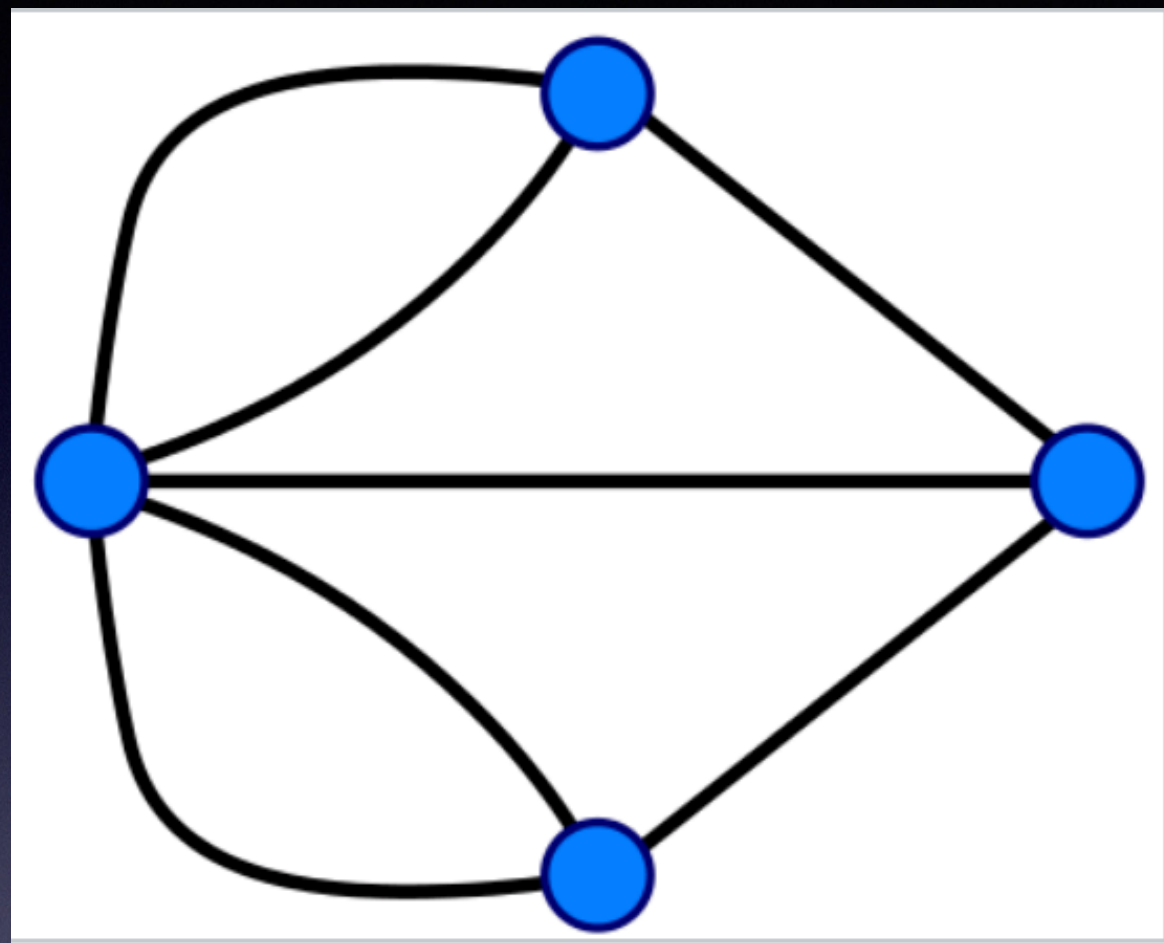


一筆書きできますか？

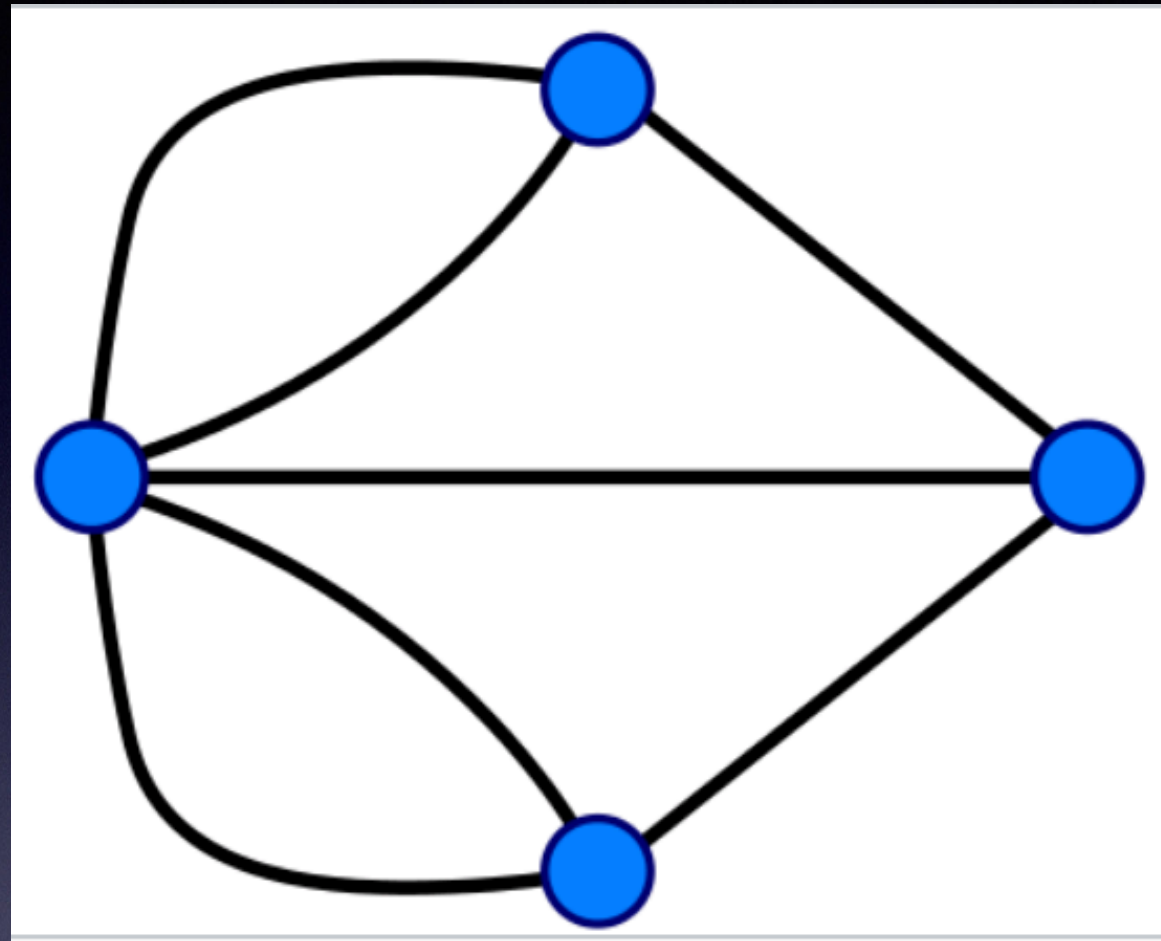


一筆書きできますか？

僕はできます
誰もができます



一筆書きできますか？



一筆書きできますか？

僕はできません

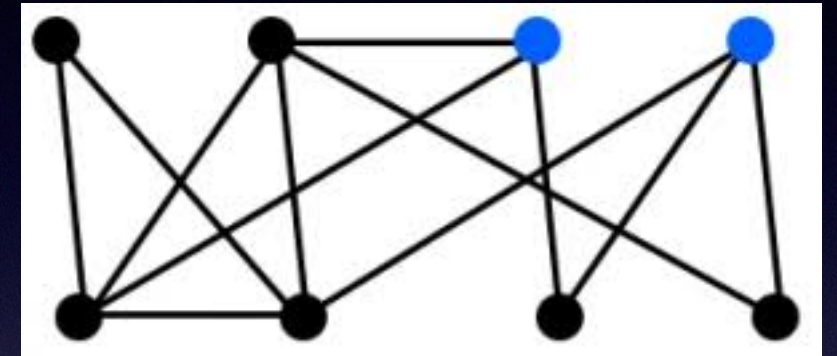
誰もできません

オイラー(閉)路

- ・ 一筆書き(路)のことです
 - ・ オイラーグラフ
 - ・ 始点と終点が同じ(閉路)一筆書きができるグラフ
 - ・ 準オイラーグラフ
 - ・ 始点と終点が違う一筆書きができるグラフ
- ・ 情エレ2年後期の情報数学で学べます(多分)

(準)オイラーグラフの条件

- ・ 無向グラフ
 - ・ 連結 && 全ての点の次数が偶数
 - ・ (準)連結 && 2点だけ点の次数が奇数
- ・ 有向グラフ
 - ・ 連結 && 点の相対出次数(出る辺の総和-入る辺の総和)が0
 - ・ (準)連結 && 一点ずつだけ相対出次数が1, -1他は0
- ・ 上を満たすならオイラー閉路、下ならオイラー路があります



準オイラーグラフ

オイラーグラフ判定の問題

- ・ AOJ 0225 こぶたぬきつねこ

- ・ <https://onlinejudge.u-aizu.ac.jp/problems/0225>

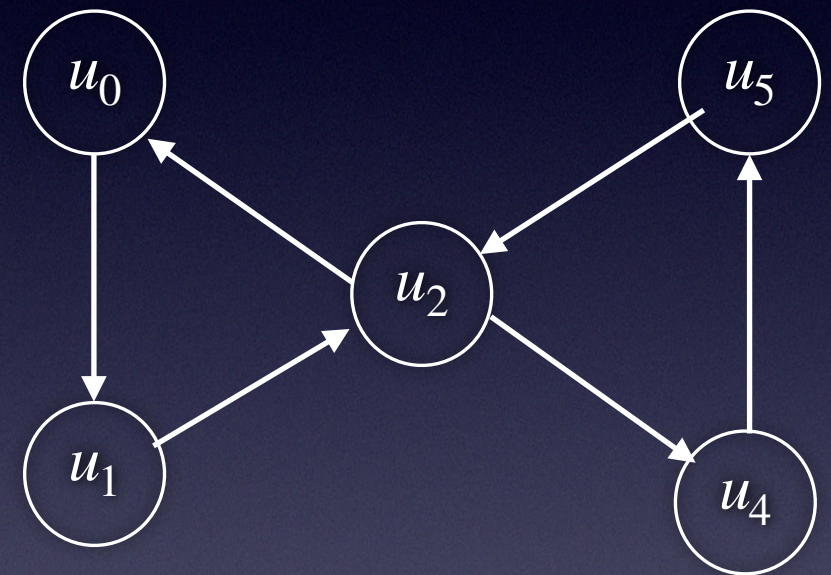
- ・ char(u8)を点と

- ・ ぼくのRust実装

- ・ <https://onlinejudge.u-aizu.ac.jp/status/users/asako0912/submissions/1/0225/judge/4069285/Rust>

Hierholzerのアルゴリズム

- ・ 入力: **オイラーグラフ**(有向)
- ・ 出力: オイラー(閉)路
- ・ $O(|E|)$, E …辺の数
- ・ 分かる人にはDFS(深さ優先探索)



Hierholzerのアルゴリズム

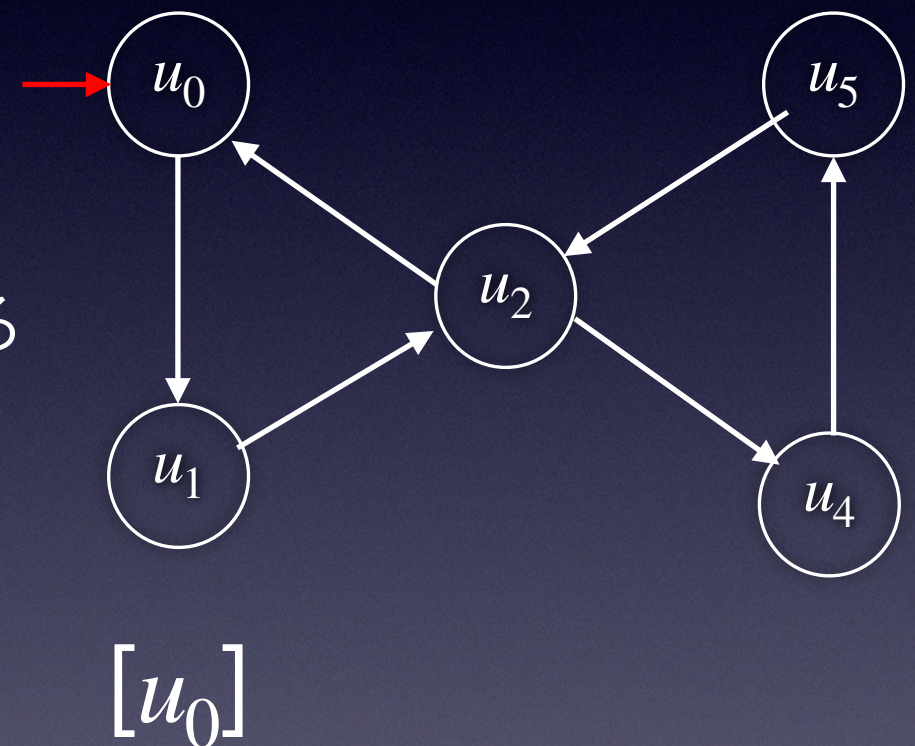
・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする

[]



Hierholzerのアルゴリズム

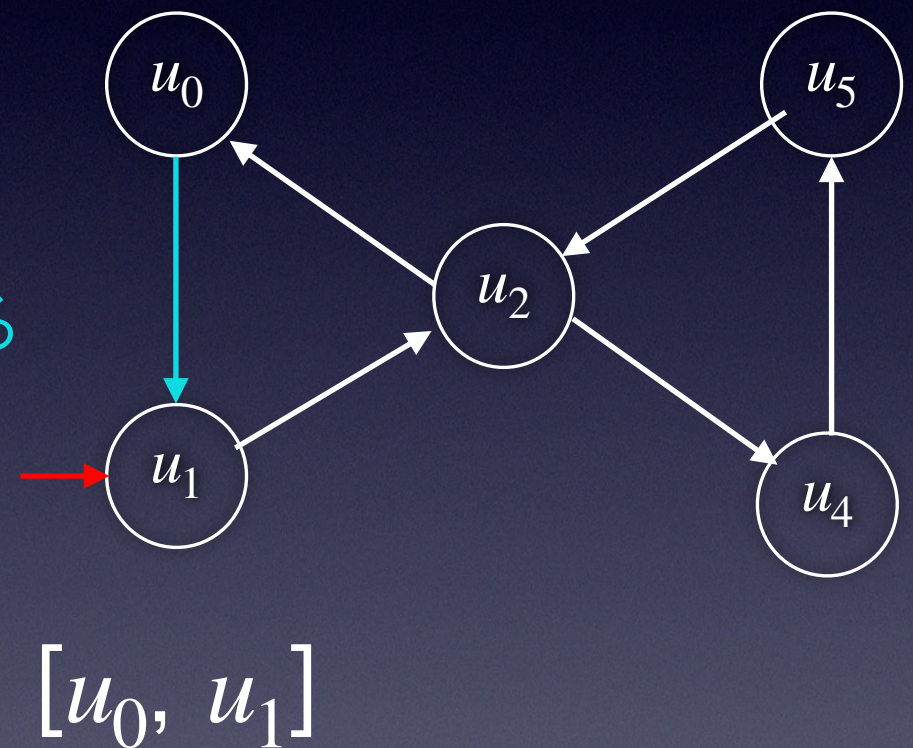
- ・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする

[]



Hierholzerのアルゴリズム

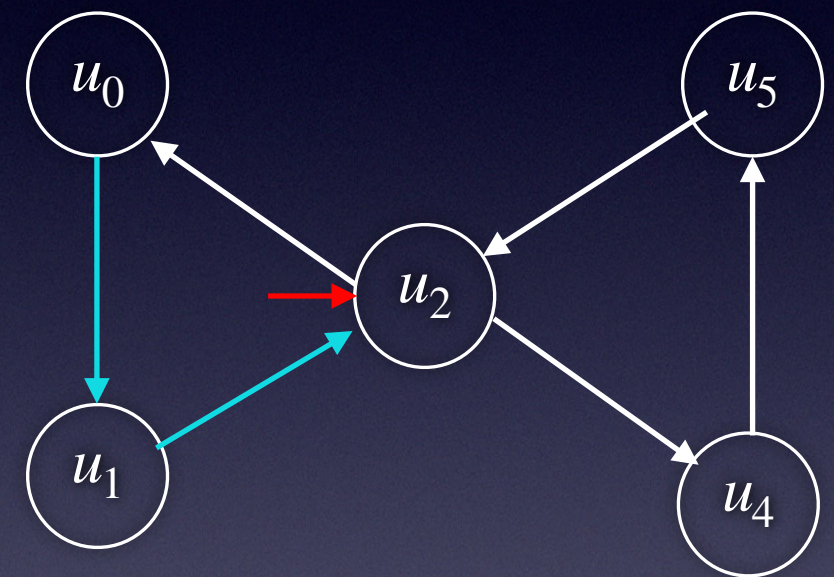
- ・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする

[]



$[u_0, u_1, u_2]$

Hierholzerのアルゴリズム

- ・ おおまかにしたいこと

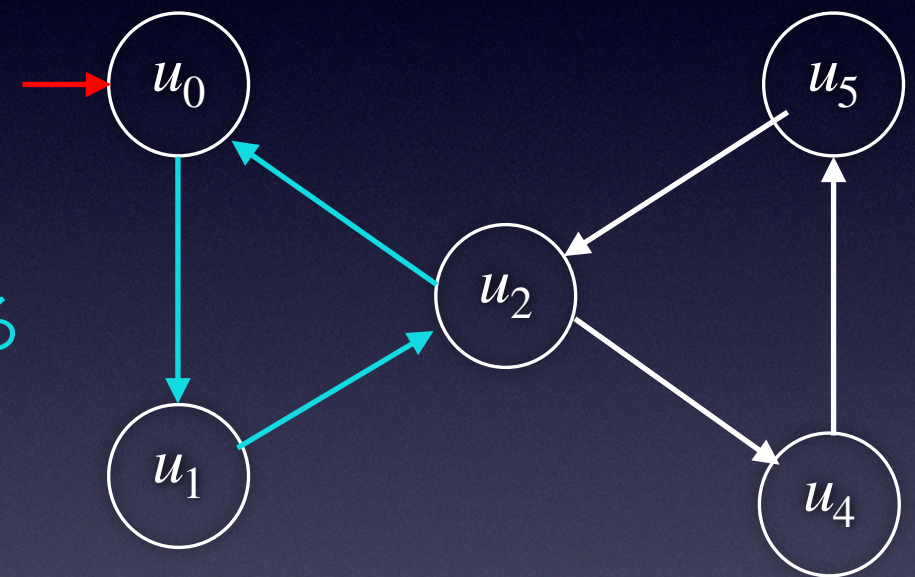
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_0]$

2. 最後に始点をpushしてreverseする

[]



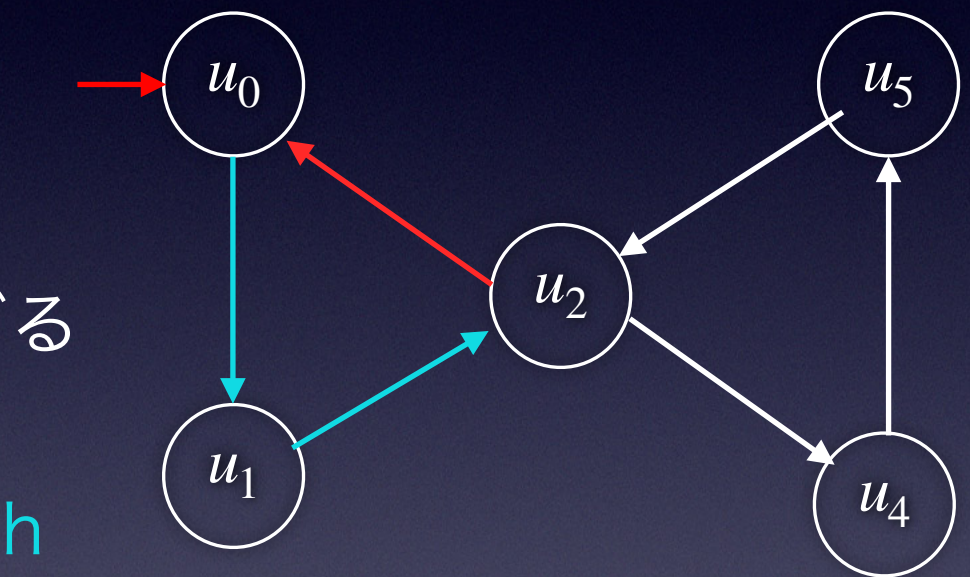
Hierholzerのアルゴリズム

u_0 からは辺がない

・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る



$[u_0, u_1, u_2]$

2. 最後に始点をpushしてreverseする

$[u_0]$

Hierholzerのアルゴリズム

u_2 からは辺があるのでたどる

・ おおまかにしたいこと

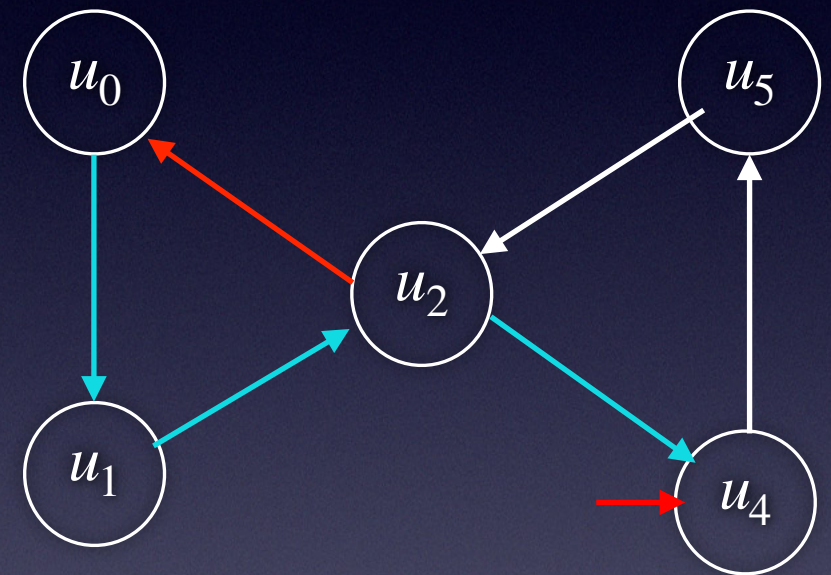
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_4]$

2. 最後に始点をpushしてreverseする

$[u_0]$



Hierholzerのアルゴリズム

・ おおまかにしたいこと

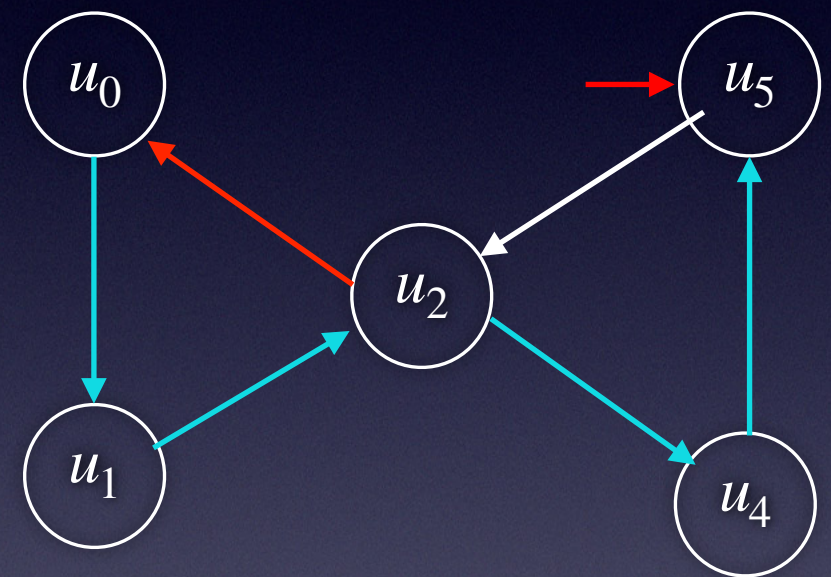
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_4, u_5]$

2. 最後に始点をpushしてreverseする

$[u_0]$



Hierholzerのアルゴリズム

次数の条件から必ず(閉)路がある

・ おおまかにしたいこと

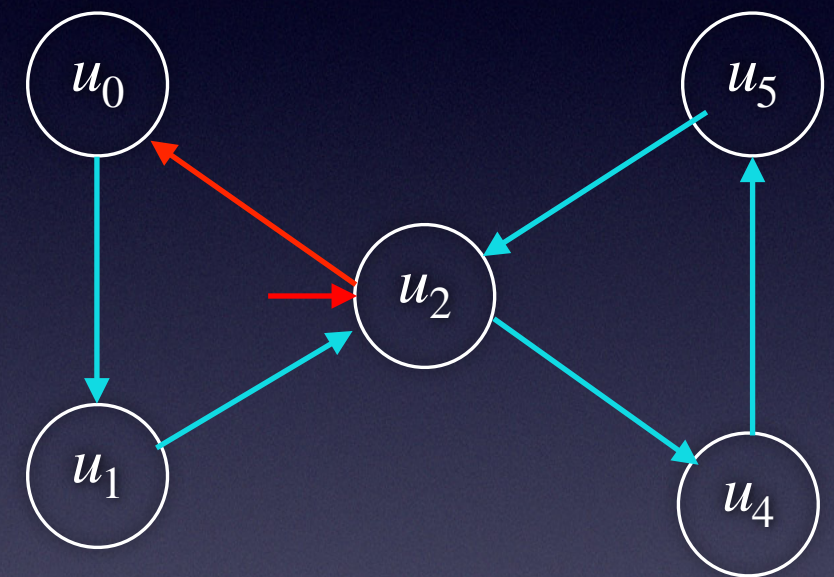
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_4, u_5, u_2]$

2. 最後に始点をpushしてreverseする

$[u_0]$



Hierholzerのアルゴリズム

・ おおまかにしたいこと

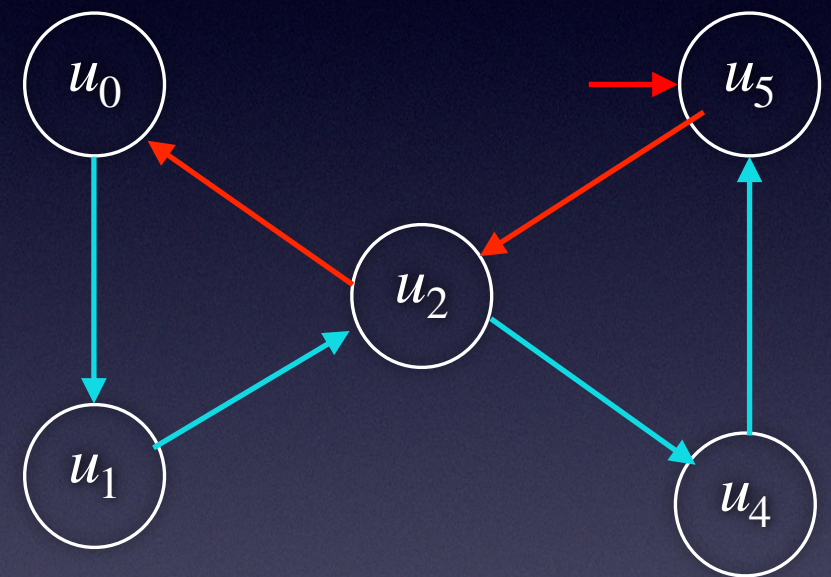
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_4, u_5]$

2. 最後に始点をpushしてreverseする

$[u_0, u_2]$



Hierholzerのアルゴリズム

- ・ おおまかにしたいこと

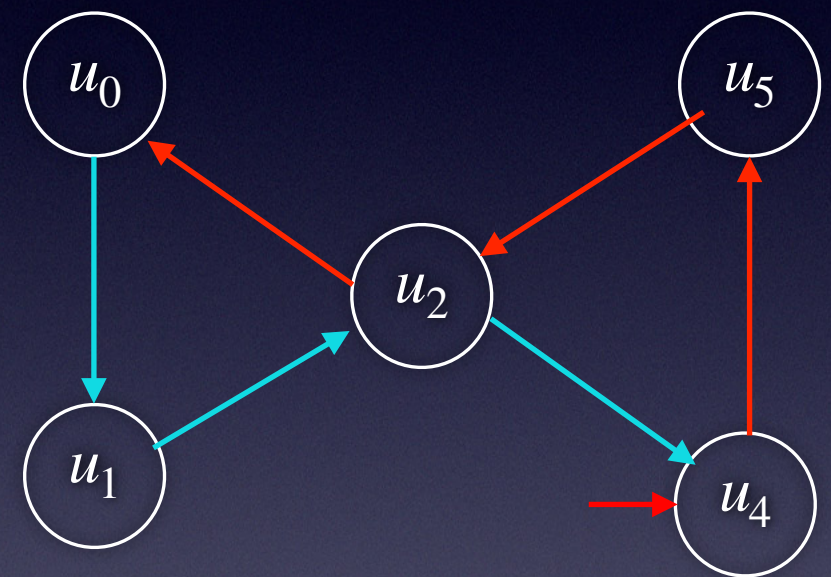
1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

$[u_0, u_1, u_2, u_4]$

2. 最後に始点をpushしてreverseする

$[u_0, u_2, u_5]$



Hierholzerのアルゴリズム

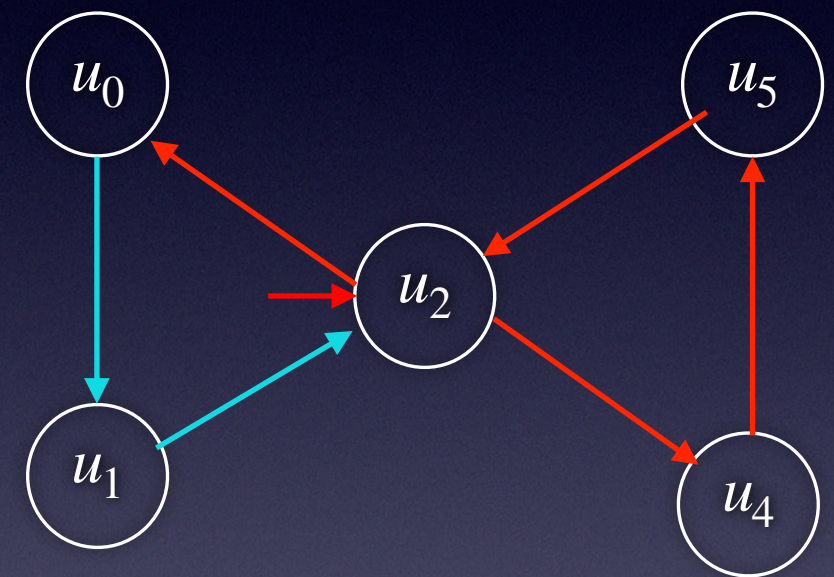
・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする

$[u_0, u_2, u_5, u_4]$



$[u_0, u_1, u_2]$

Hierholzerのアルゴリズム

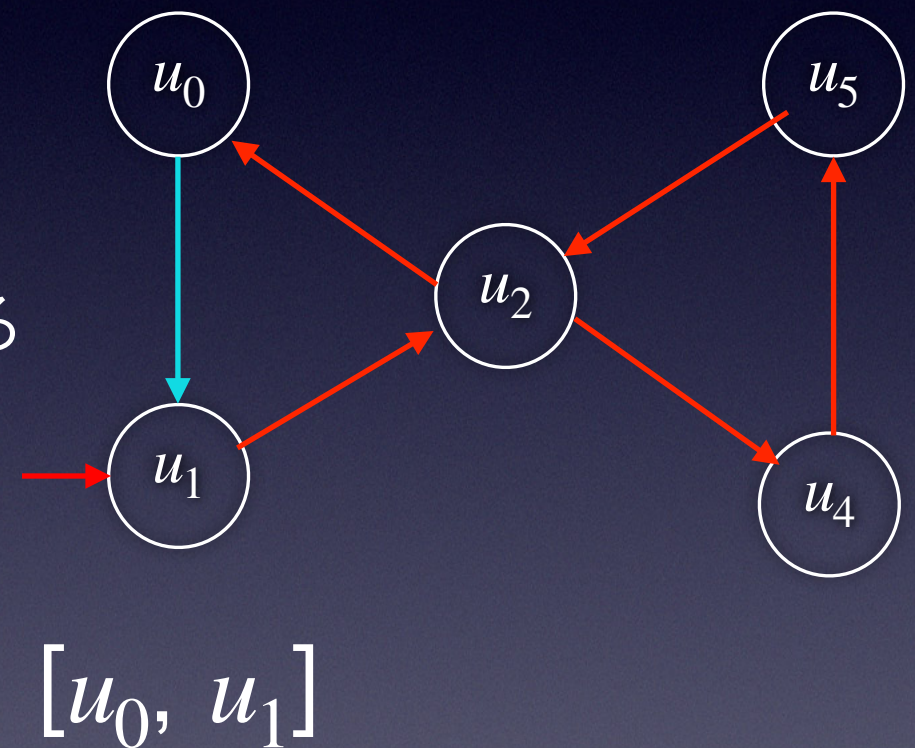
- ・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpushして一辺戻る

2. 最後に始点をpushしてreverseする

$[u_0, u_2, u_5, u_4, u_2]$



Hierholzerのアルゴリズム

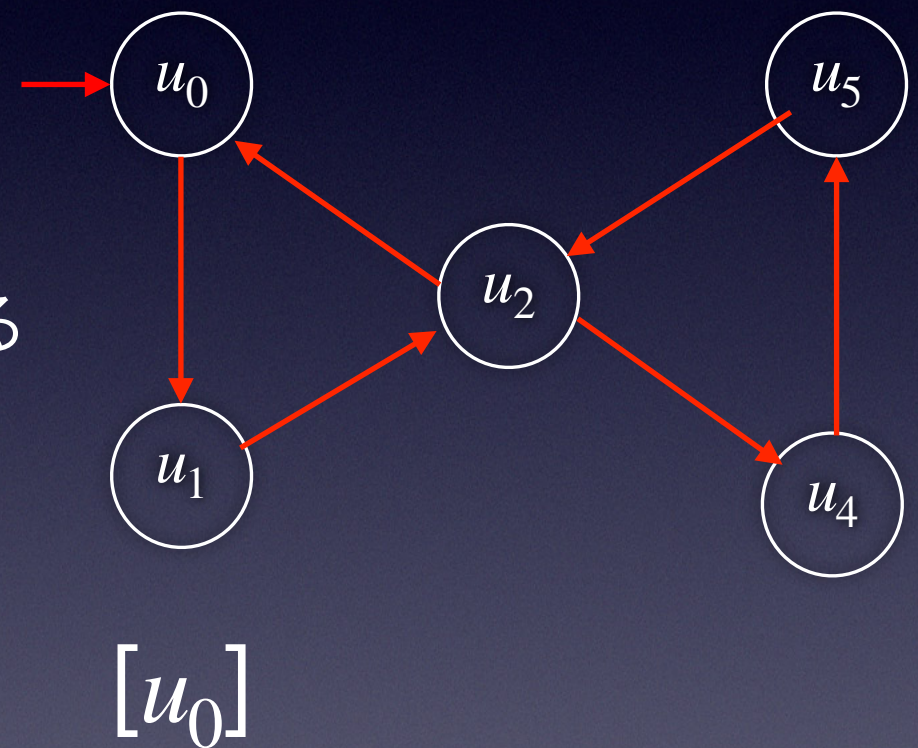
・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする

$[u_0, u_2, u_5, u_4, u_2, u_1]$



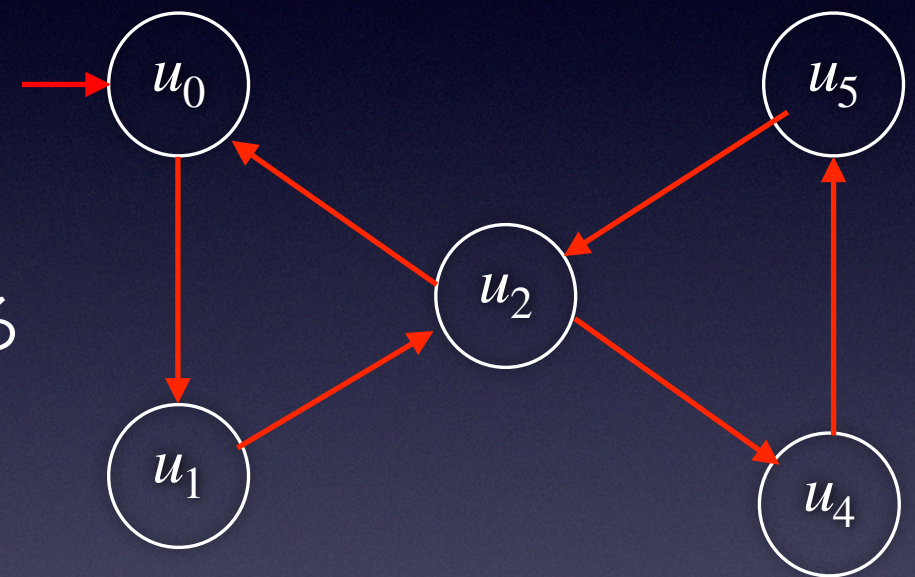
Hierholzerのアルゴリズム

・ おおまかにしたいこと

1. 始点から辺がなくなるまで辺をたどる

1. 辺がなかったら今いる点をpush
して一辺戻る

2. 最後に始点をpushしてreverseする



$[u_0, u_1, u_2, u_4, u_5, u_2, u_0]$ ← オイラー閉路(辿る点)

(有向)オイラー路の問題

- ・ CF288 D. Tanya and Password

- ・ <http://codeforces.com/contest/508/problem/D>

- ・ 点と辺の設定は前の問題と似たような感じ

- ・ ぼくのRust実装

- ・ <http://codeforces.com/contest/508/submission/67508623>

おしながき

- ・ 自己紹介
- ・ オイラー路
- ・ Rustを使ってみませんか
- ・ おまけ: 明日から使えるbit abs min

Rustを使ってみませんか

- ・ 僕はRustをひっそりと書いています
 - ・ Rustってなんですか
 - ・ 何ができるんですか
 - ・ 使って何が嬉しいんですか

Rustってなんですか

- ・ 僕の口から言うのは畏れ多いので公式をどうぞ
- ・ <https://www.rust-lang.org/>

何ができるんですか

- ・ ブラウザが書けます(僕は書けません)
 - ・ Let's build a browser engine!
 - ・ <https://limpet.net/mbrubeck/2014/08/08/toy-layout-engine-1.html>
- ・ Rubyが書けます(僕は
 - ・ <https://qiita.com/sisshiki1969/items/3d25aa81a376eee2e7c2>
- ・ 実行プログラム作成基盤が書けます
 - ・ <https://drumato.hatenablog.com/entry/2019/12/08/0000000>

何が嬉しいの

- ・ コンパイラが参照やスコープについて丁寧に教えてくれる
- ・ いろんな言語パラダイムの機能があるので詳しくなれる
- ・ 型意識が高くなる
- ・ 強い人達が低レイヤーな面白いモノを作って公開している



おしながき

- ・ 自己紹介
- ・ オイラー路
- ・ Rustを使ってみませんか
- ・ おまけ: 明日から使えるbit abs min

if文を使わずに
abs(signed int a)
を書けますか

if文を使わずに
abs(signed int a)
を書けますか

ヒント: 2の補数表現の知識と算術右シフトを使います

abs

```
fn abs(a: i64) -> i64 {  
    let b = a >> 63;  
    return (b ^ a) - b;  
}
```

- ・ $a < 0$ なら $b = -1 = 0x1111\cdots111$
 - ・ $(b \oplus a) - b = !a + 1$ (2の補数表現の符号反転の処理)
- ・ $a \geq 0$ なら $b = 0 = 0x0000\cdots000$
 - ・ $(b \oplus a) - b = a$
- ・ Rustはsignedに対する右シフトは算術
- ・ C/C++は右シフトが算術/論理は処理系定義(まれにだめかも)

if文を使わずに
min(signed int a, signed int b)
を書けますか

if文を使わずに
min(signed int a, signed int b)
を書けますか

ヒント: if文は使わないけど $a < b$ は使います(?)
この時大泉さんによって単純に書けることが示された

$\text{return } (a < b) * a + (1 - (a < b)) * b$

if文を使わずに
min(signed int a, signed int b)
を書けますか

ヒント: if文は使わないけど $a < b$ は使います(?)
この時大泉さんによって単純に書けることが示された

では、
掛け算(*)を使うこと
($a < b$)を二回以上使うこと
を禁止します(苦しい)

min(max)

```
fn min(a: i64, b: i64) -> i64 {  
    return b ^ ((a ^ b) & -((a < b) as i64));  
}
```

- ・ C, C++ならboolはintでキャスト (as i64) がいらないので美しい
- ・ $a < b$ なら $-(a < b) = -1 = 0x1111 \dots 111$
 - ・ $b \oplus ((a \oplus b) \wedge -1) = a. (\because (a \oplus b) \wedge -1 = a \oplus b, \quad b \oplus (a \oplus b) = b \oplus b \oplus a = 0 \oplus a = a)$
- ・ $a \geq b$ なら $-(a < b) = 0 = 0x0000 \dots 000$
 - ・ $b \oplus ((a \oplus b) \wedge 0) = b. (\because (a \oplus b) \wedge 0 = 0, \quad b \oplus 0 = b)$

参考

- ・ はまやんはまやんさんのオイラー路のまとめ

- ・ <https://www.hamayanhamayan.com/entry/2016/10/05/101107>

- ・ Spaghetti Source 有向オイラー路

- ・ http://www.prefield.com/algorithm/graph/directed_euler_path.html

- ・ Hierholzer's algorithm (GeeksforGeeks)

- ・ <https://www.geeksforgeeks.org/hierholzers-algorithm-directed-graph/>

- ・ MIT OpenCourseWare 3. Bit Hacks

- ・ https://www.youtube.com/watch?v=ZusiKXcz_ac&list=PLUI4u3cNGP63VIBQVWguXxZZi0566y7Wf&index=3

- ・ コンピュータにおける整数の表現(低レイヤを知りたい人のための…)

- ・ https://www.sigbus.info/compilerbook#twos_complement

- ・ Rust(Wasm)でNES(ファミコン)エミュレーターが書けます(b

- ・ <https://blog.bokuweb.me/entry/2018/02/08/101522>

いいわけ

- ・ あまり見栄えするものを作っていないので雑学です
- ・ オイラー路
 - ・ 授業(数学科)でやった
 - ・ 実装は自分でやった
 - ・ 競プロブームに乗った
- ・ Rust
 - ・ 最近趣味で書いている
 - ・ 仲間が欲しい
- ・ おまけ
 - ・ 最近知った