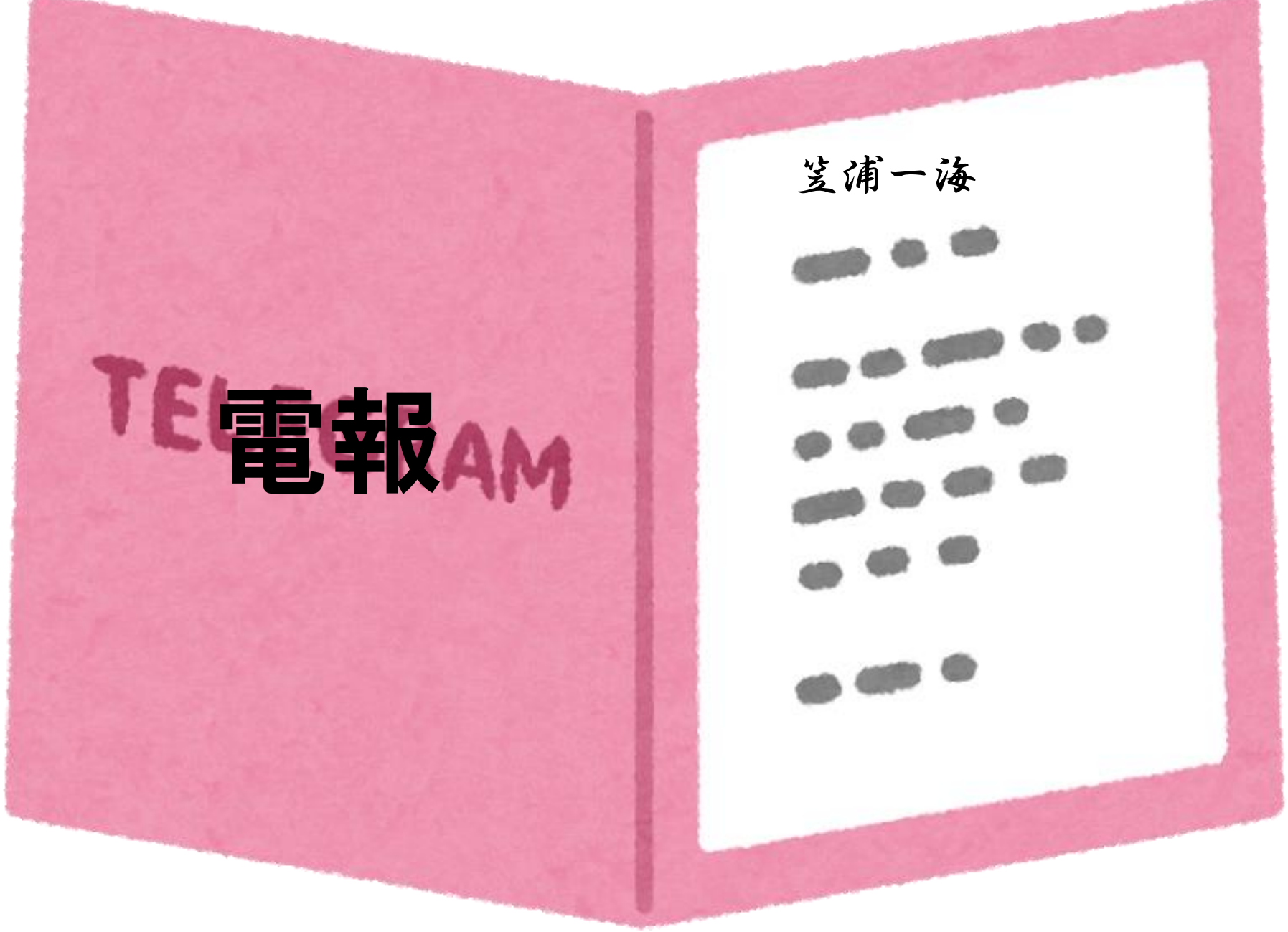

TELEGRAM
電報
笠浦一海

問題概要

- 全頂点の出次数が1の有向グラフがある(Functional Graph)
- コストを払えば辺の行先を変えられる
- 全体を強連結にしたい
- 必要な最小のコストは？

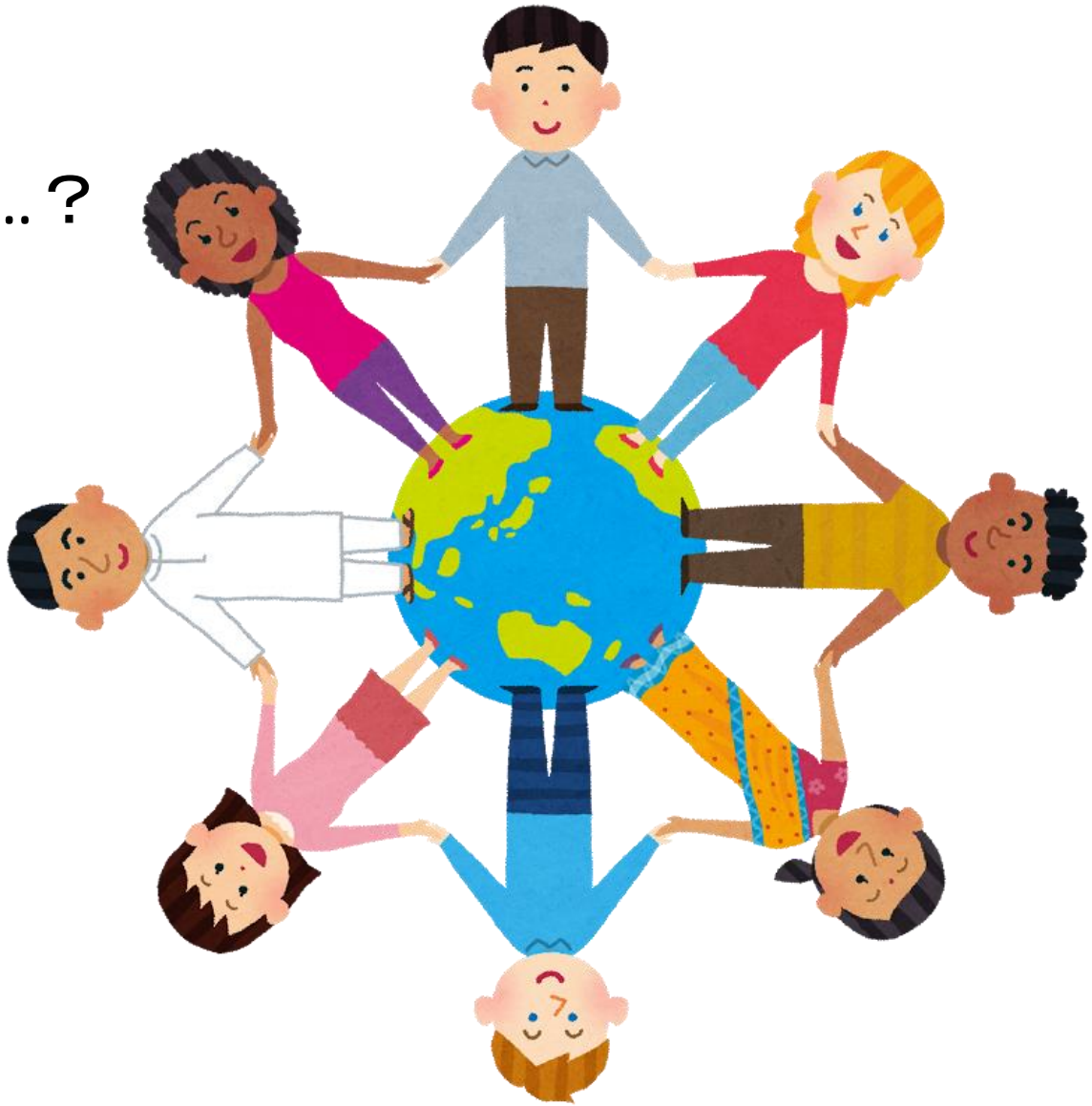
超重要考察

- 強連結なFunctional Graphとは……？

超重要考察

- 強連結なFunctional Graphとは……？

•輪



全探索解法

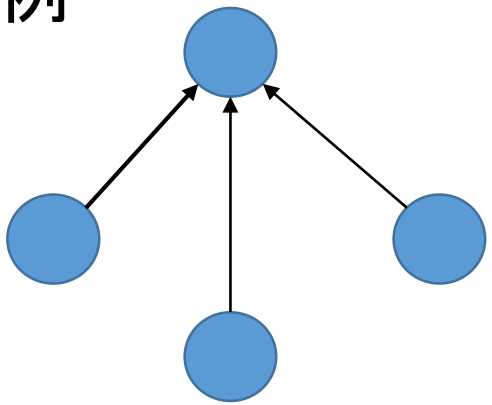
- N 個の頂点を輪状に並べる方法は (N-1)! 通り
- それぞれの並べ方について、その状態にするのにかかるコストを求める
- 行き先を変更する必要のある辺を調べてコストを足すだけ O(N)
- 全体で O(N!)
- Subtask 1 が解けた

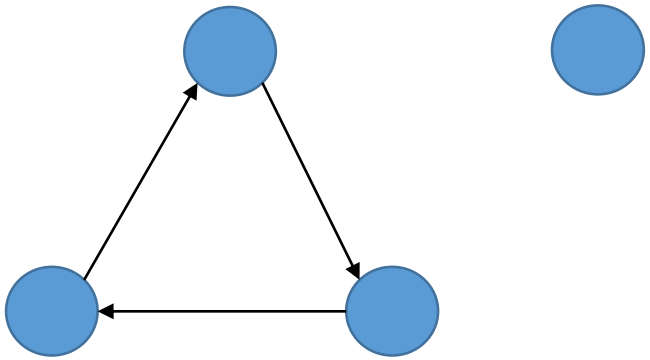
重要考察

- コストは「どの辺を行先を変更せずに残すか」のみによって決まる
- 変更にかかるコスト＝（全辺のコストの合計）ー（残す辺のコストの合計）
- ありえる「残す辺の集合」のうちコストの合計が最大のものを求めればよい。

「残す辺の集合」としてあり得るものは？

- サイクルグラフの部分グラフになっていればよい
- 駄目な例





- 入次数が2以上の頂点がある　→　**駄目**
- サイクルがある（全体がサイクルの場合を除く）　→　**駄目**

「残す辺の集合」としてあり得るものは？

- 逆に全頂点の入次数が1以下でサイクルがないとき
- （注意：出次数は常に1以下）

- いくつかのパスからなるグラフ　→　適当につなげれば輪になる

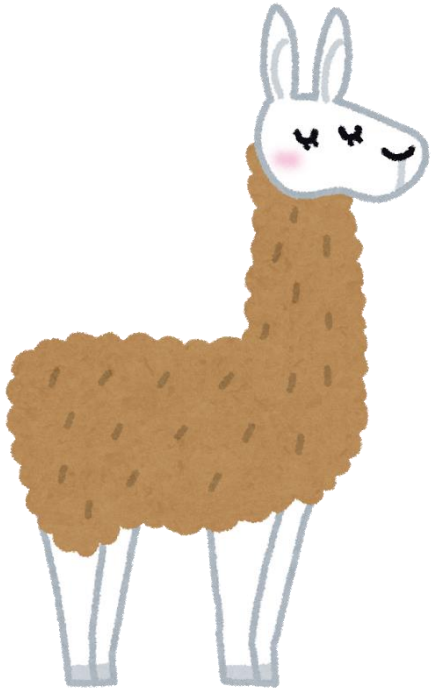
- 必要十分条件が求まった

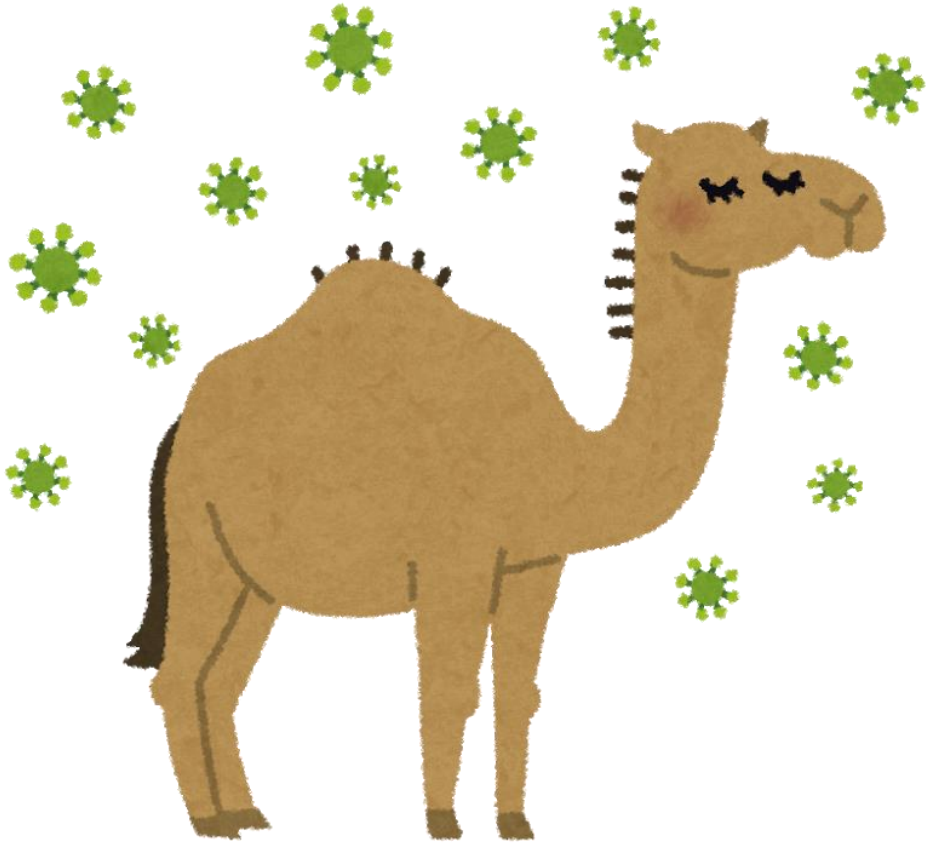
問題を言い換える

- 「全頂点の入次数が1以下で、サイクルを含まない部分グラフのうち、コストの合計が最大のものを求めよ」
- ただし初めから輪になっている場合は例外として処理
- 辺集合の部分集合の個数：2^N
- 条件を確かめる：O（N）　（実装については後述）
- 全体でO(2^N N)
- Subtask 2 が解けた
- ※bit DP でもSubtask 2 は解ける

Greedy でできないか？

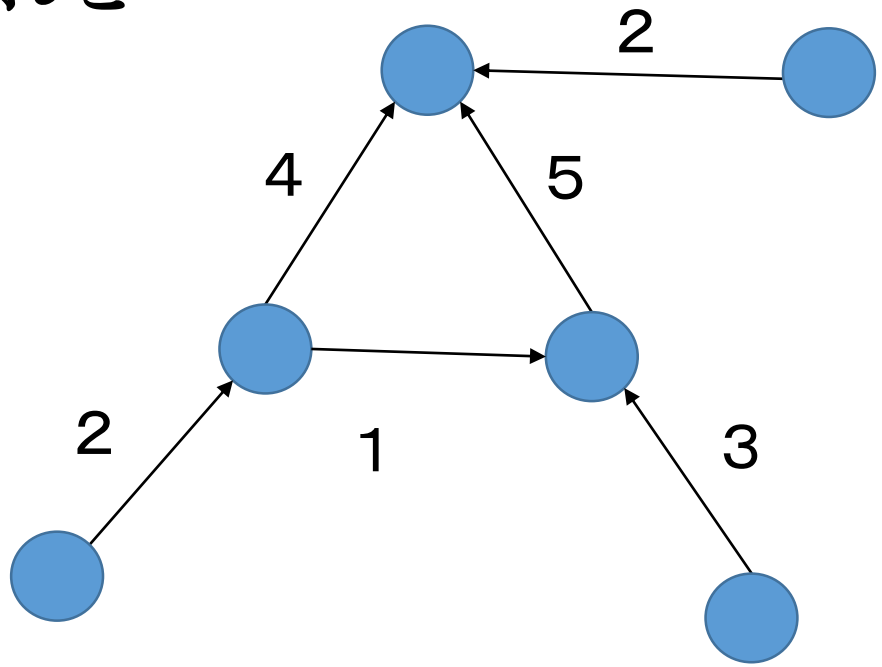
- 「サイクルをふくまない」という条件はとりあえず無視して「全頂点の入次数が1以下」だけ考える
- すると Greedy で簡単に解ける





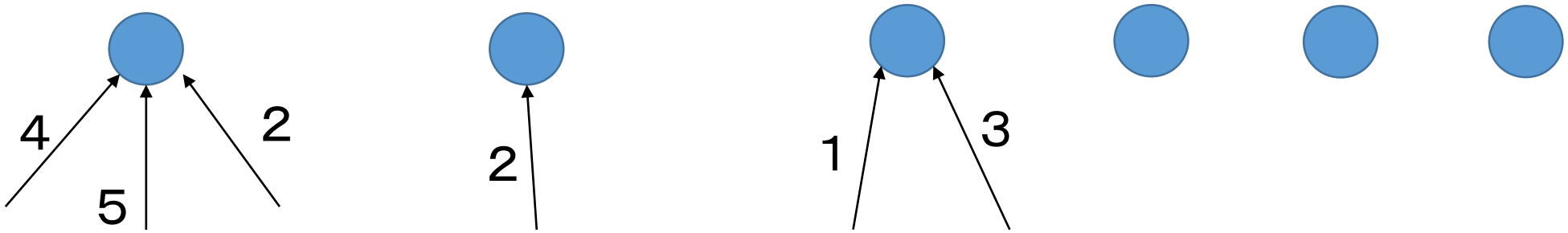
Greedy でできないか？

- 辺を行先ごとにグループ分け
- これを……



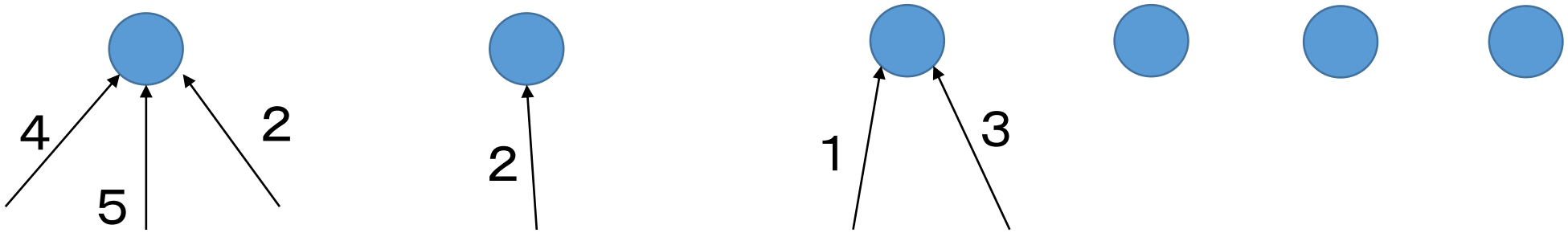
Greedy でできないか？

- 辺を行先ごとにグループ分け
- こうして……



Greedy でできないか？

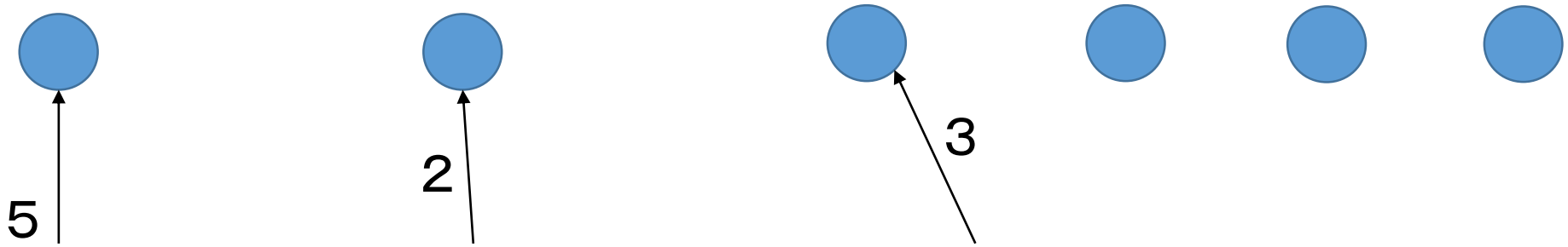
- 辺を行先ごとにグループ分け
- こうして……



- 各グループで最もコストの大きい辺をとる

Greedy でできないか？

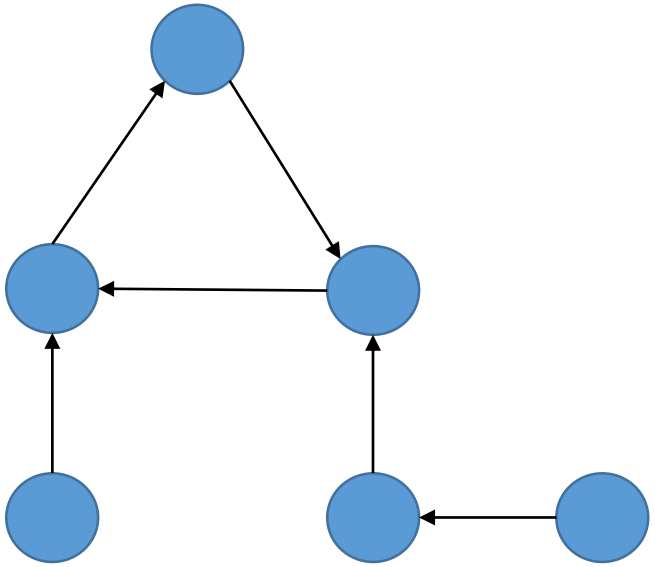
- 辺を行先ごとにグループ分け
- こうじゃ！

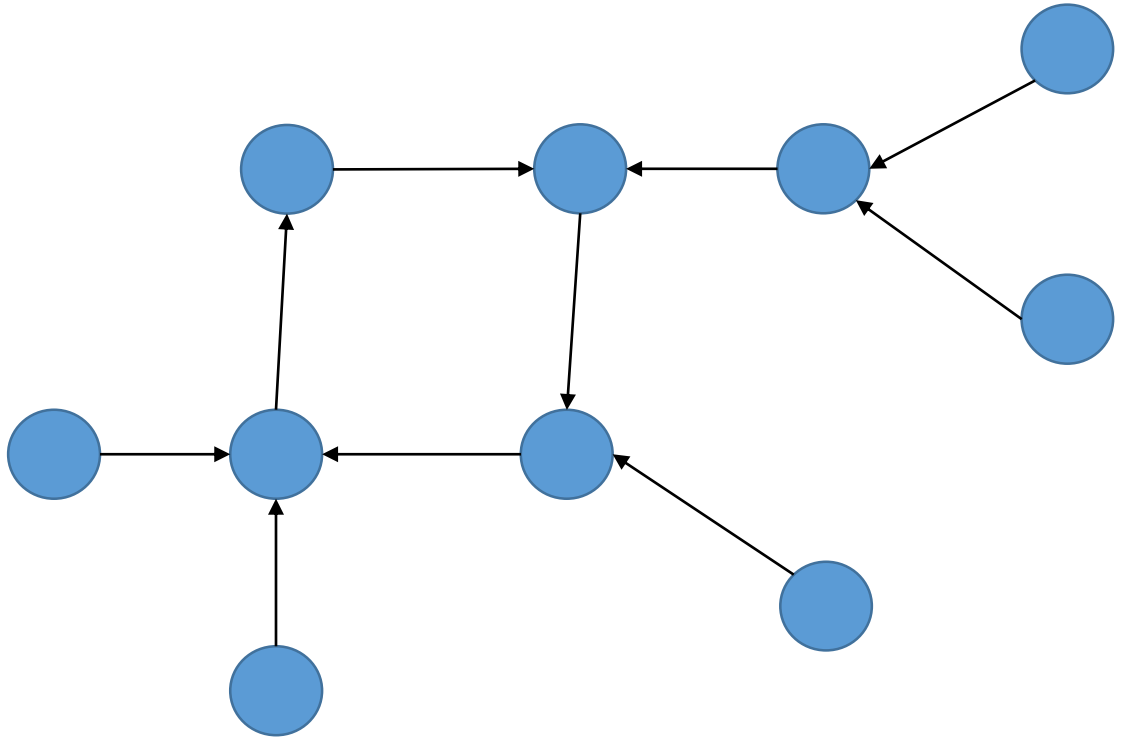


- 各グループで最もコストの大きい辺をとる

Functional Graph の形

- 「サイクルをふくまない」を考慮する
- Functional Graph における各弱連結成分は以下のような一つのサイクルに木がくっついた形である。サイクル同士が重なったりはしない。





サイクル上のどの辺を切るか？

- サイクル上の辺のうち少なくとも一つを切る（行先を変更する）必要がある
- とりあえず切る辺を全探索してみる
- 正確には各弱連結成分に対して、以下を行う

①サイクルの辺から「必ず切る辺」をひとつ選ぶ

②残りの辺に対して先ほどの Greedy を行う

③これをサイクルの各辺について試し、最良の結果を記録

サイクル上のどの辺を切るか？

- 「必ず切る辺」の選び方は高々 N 通り
- Greedy はO(N)
- 計算量はO(N^2)
- → Subtask 3 まで解ける

満点解法

- さっきの解法は明らかに無駄が多い
- 「必ず切る辺」を決めたことで影響を受けるのはその辺が属するグループの結果のみ

最大コスト:5　　最大コスト:4

- (その辺を含む最大コスト) ー (その辺以外の最大コスト)だけコストが減少

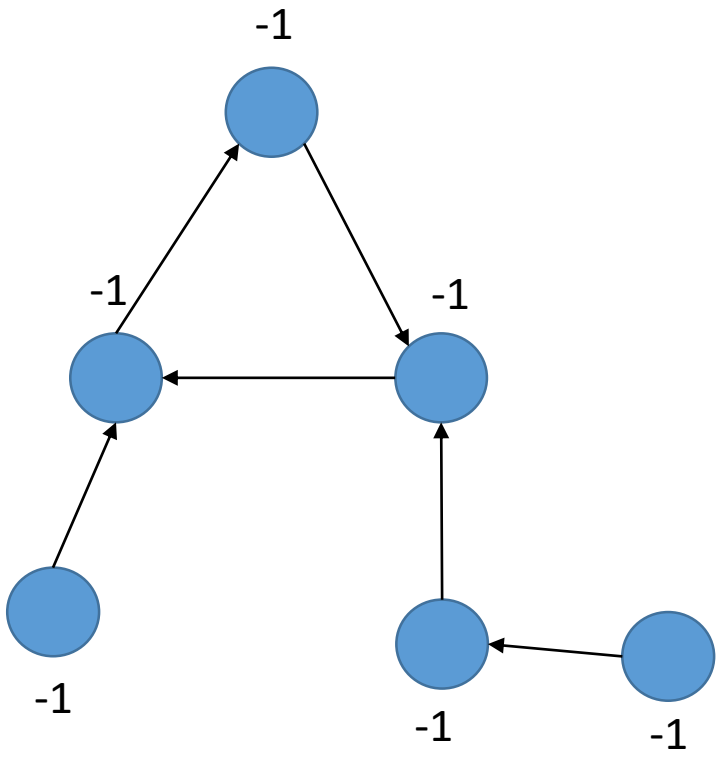
満点解法

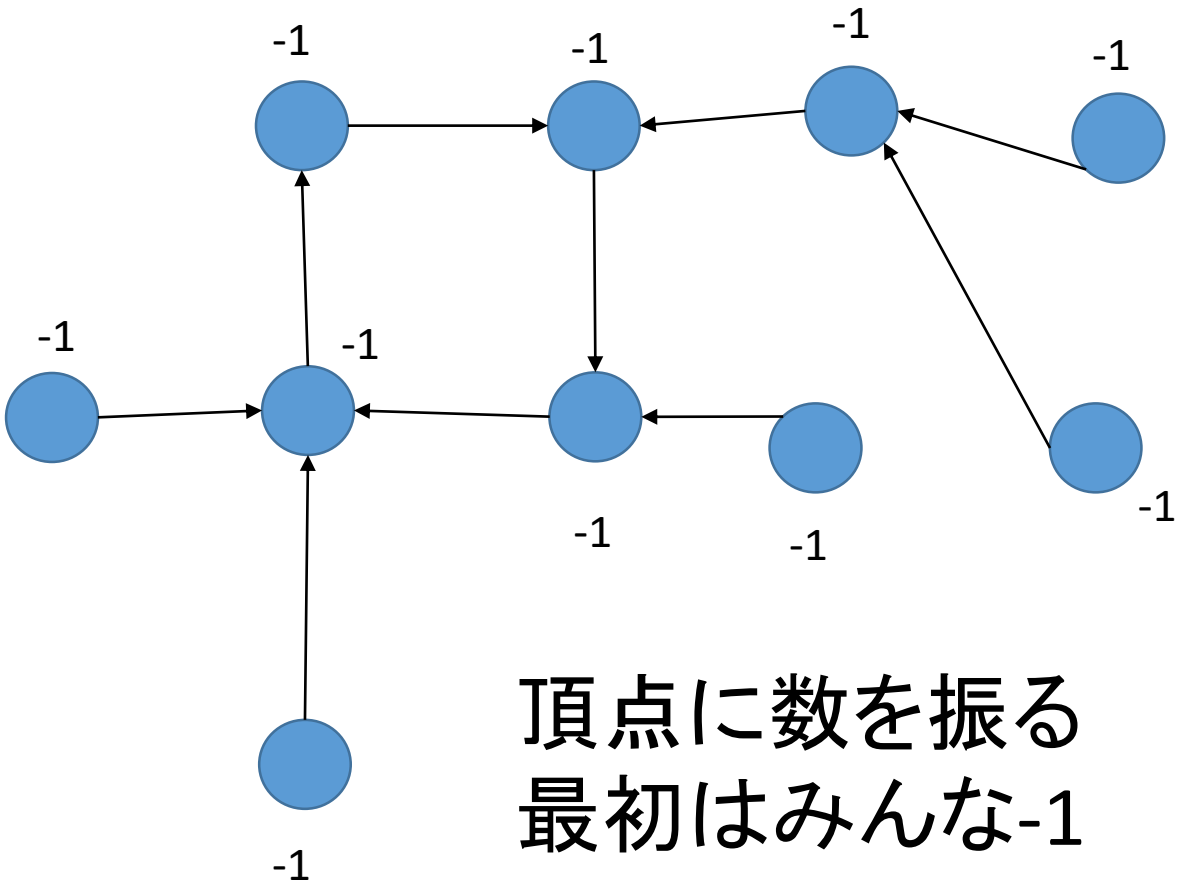
- まずサイクルの条件を無視して Greedy
- 各サイクルの各辺に対し、その辺を「必ず切る辺」に決めたときに合計コストがいくつ減少するかを求める
- 各サイクルに対し、減少コストが最小の辺を選ぶ
- → O(N) → 満点

Functional Graph の閉路検出の実装

- JOI春合宿で定期的に出題されている
- 2011年春　報告（Report）
- 2013年春　プレセント(Presents)
- 「プレセント」の解説スライドが詳しい

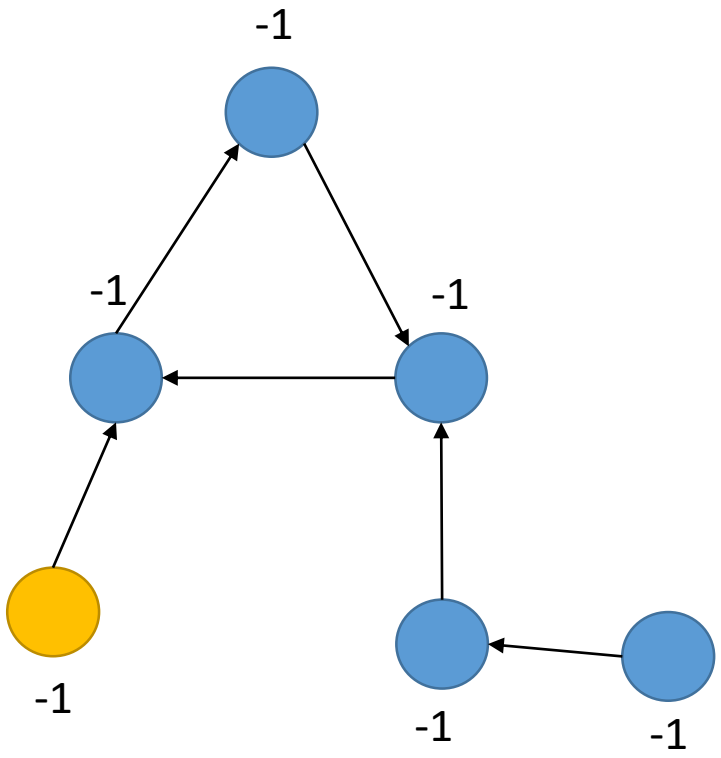
おすすめの実装

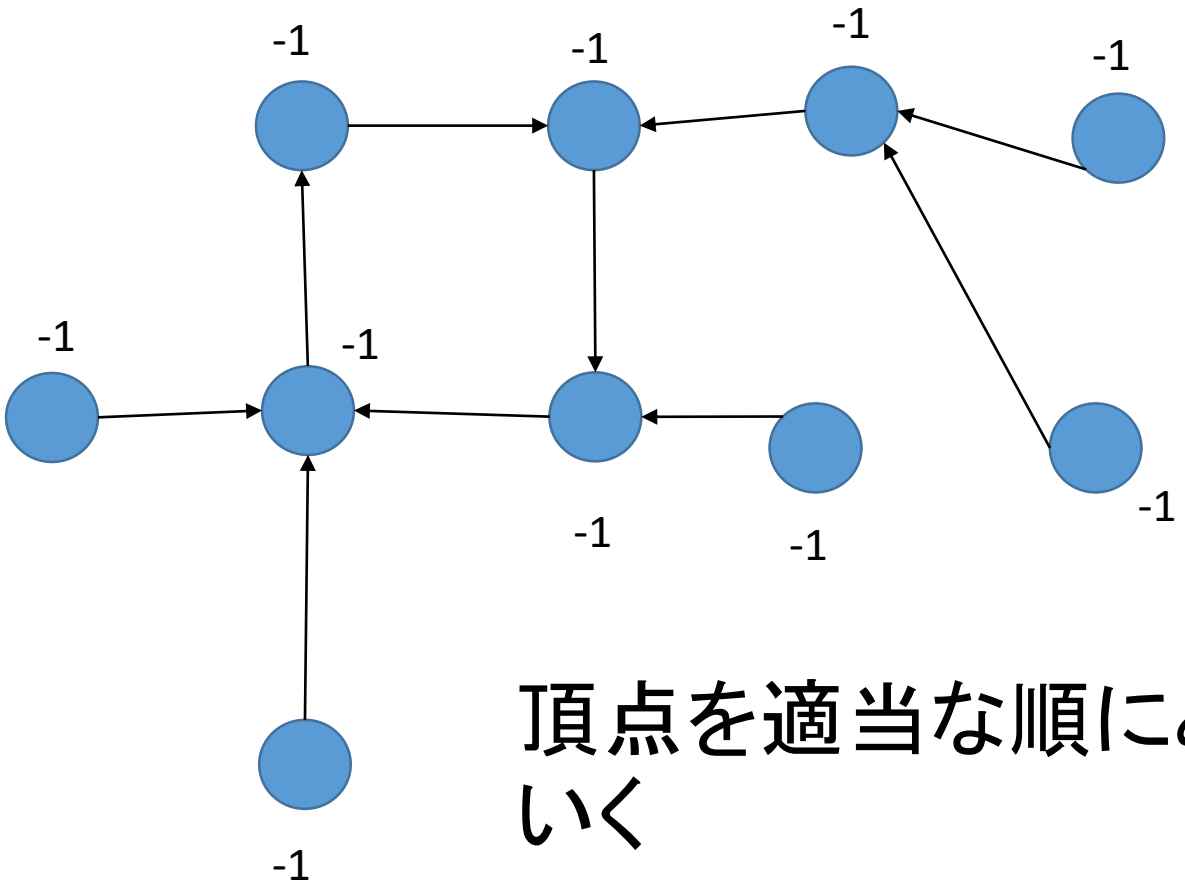




頂点に数を振る
最初はみんな-1

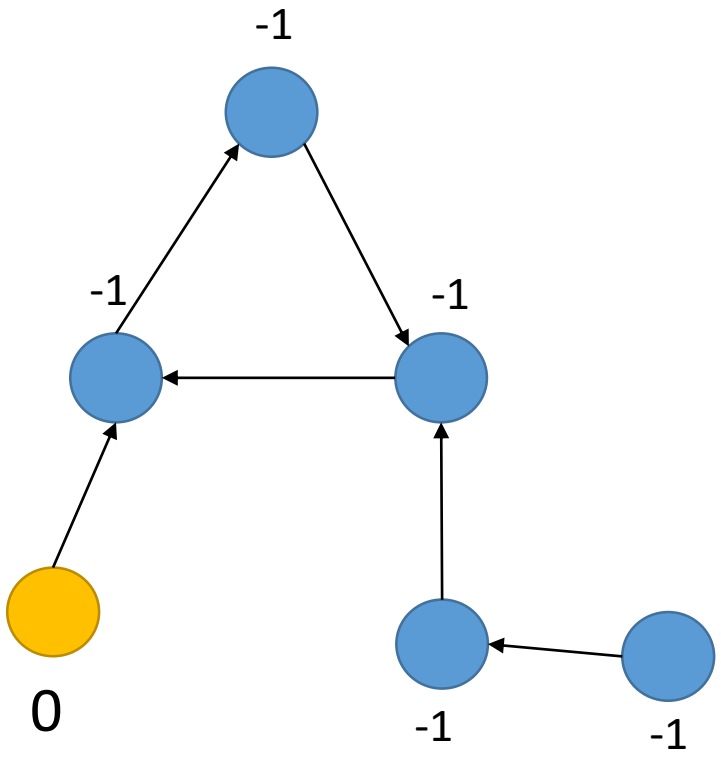
おすすめの実装

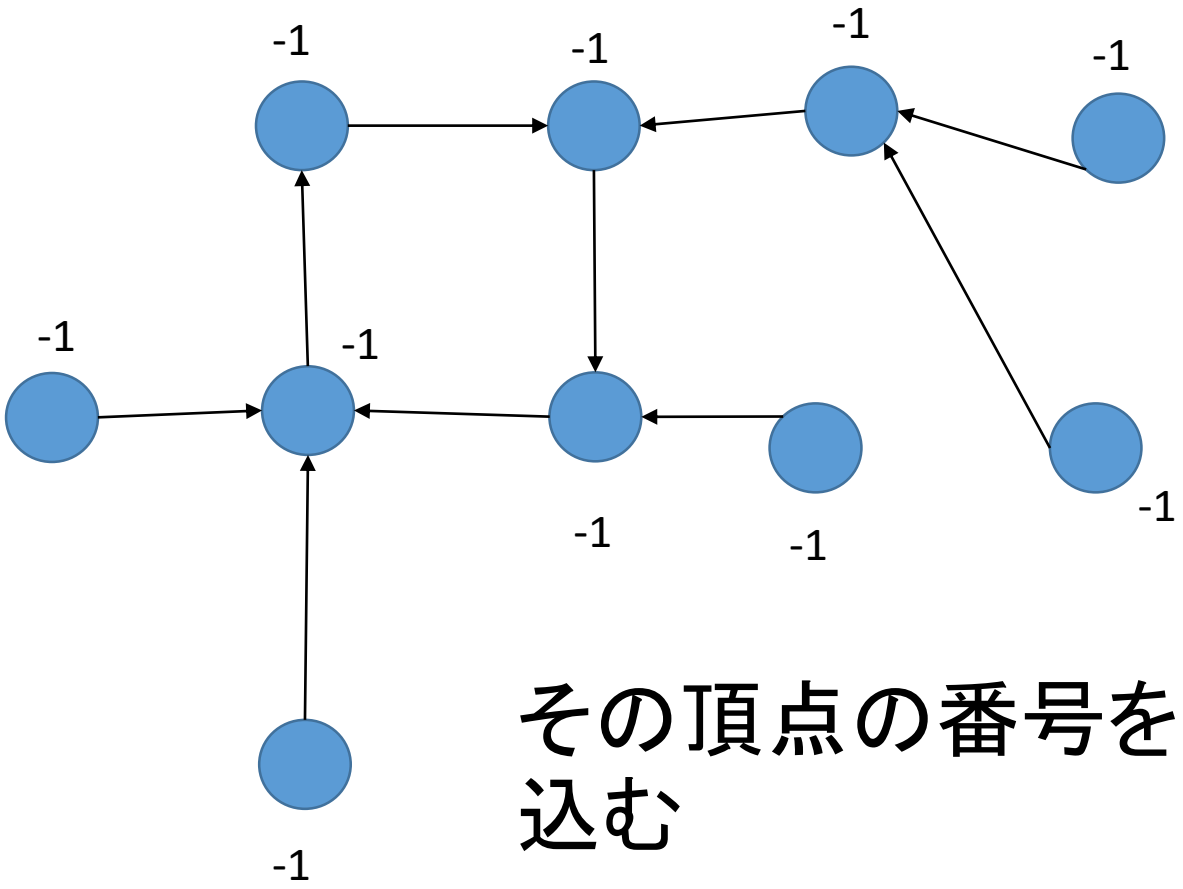




頂点を適当な順にみていく

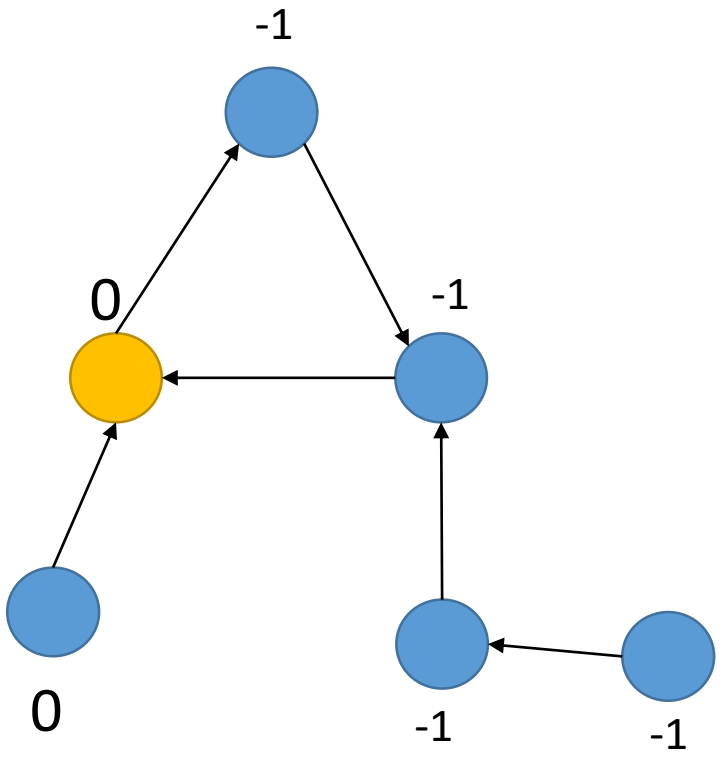
おすすめの実装

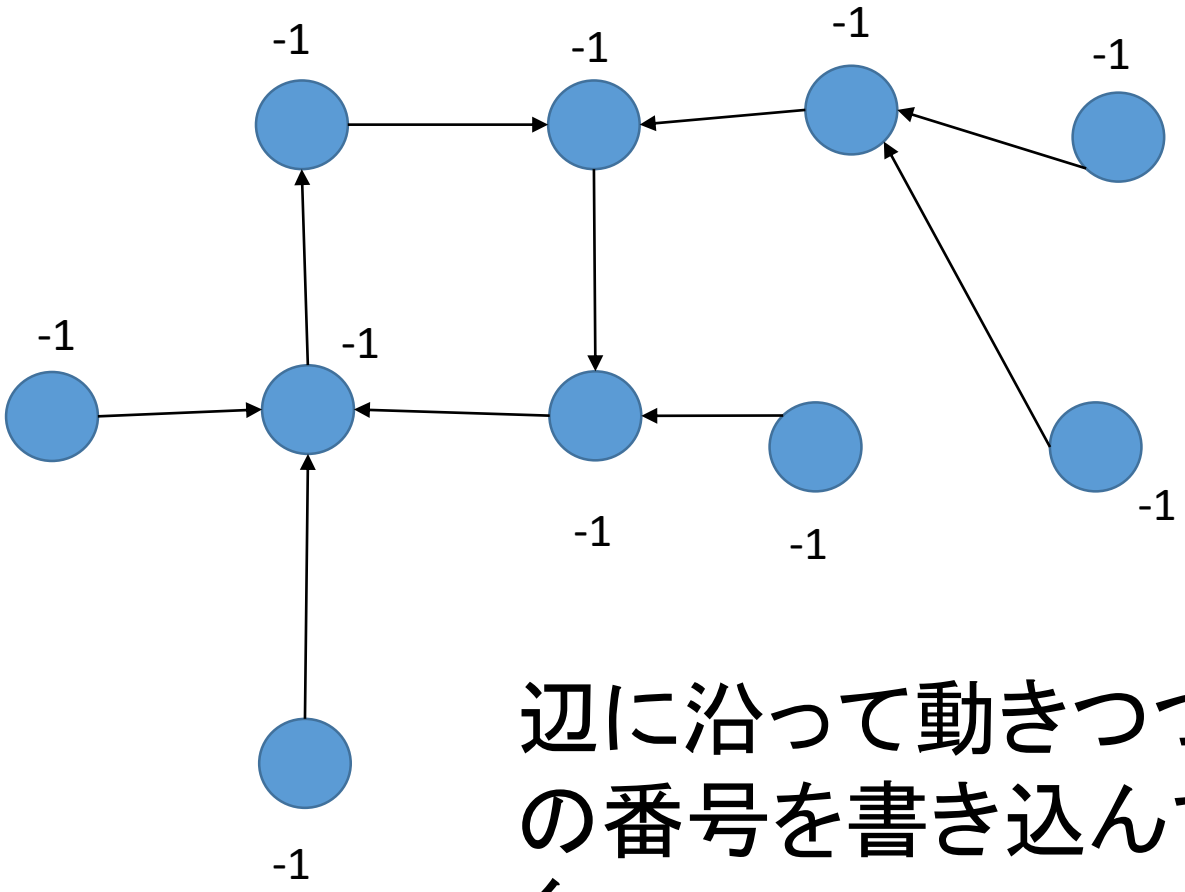




その頂点の番号を書き込む

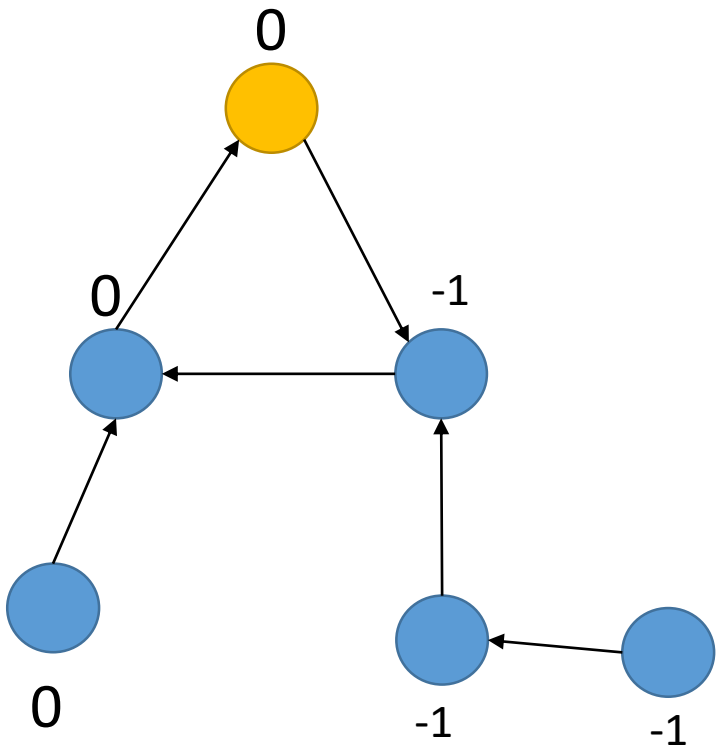
おすすめの実装

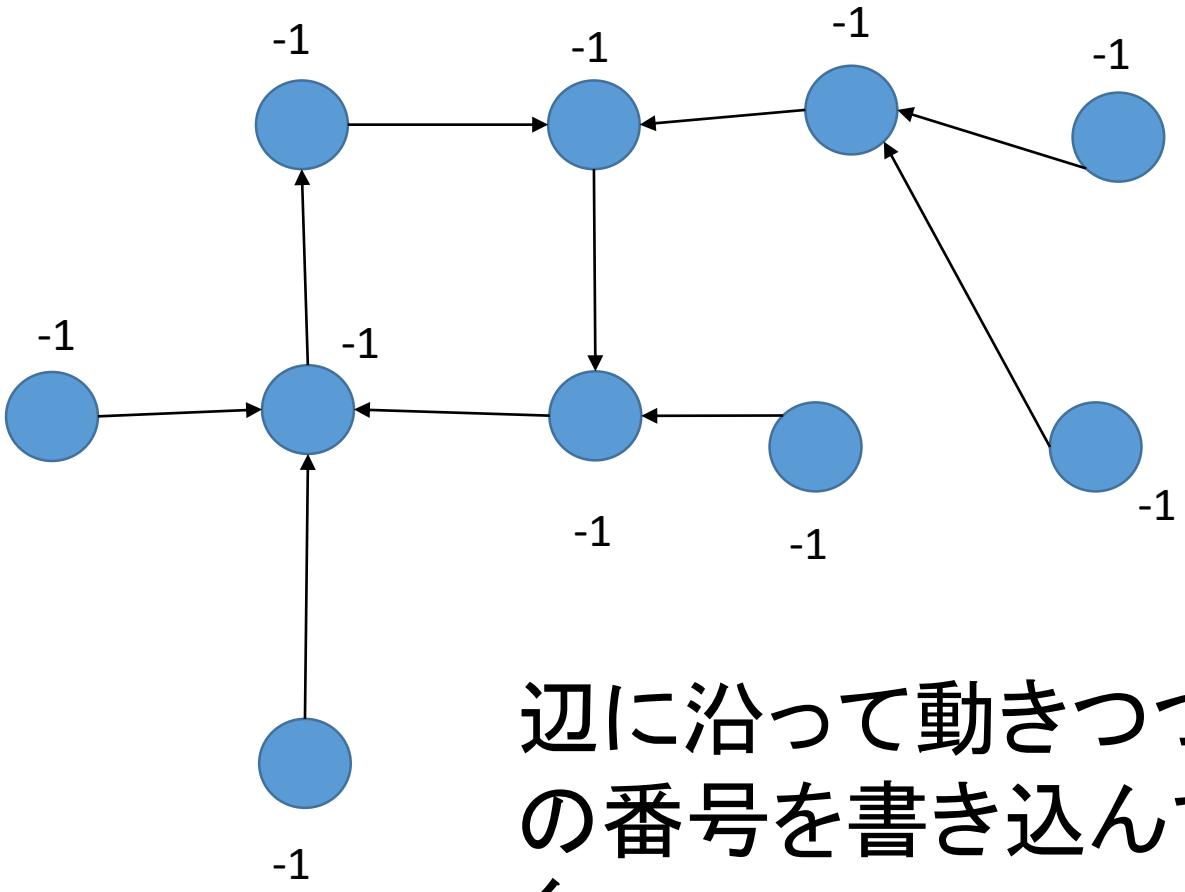




辺に沿って動きつつその番号を書き込んでいく

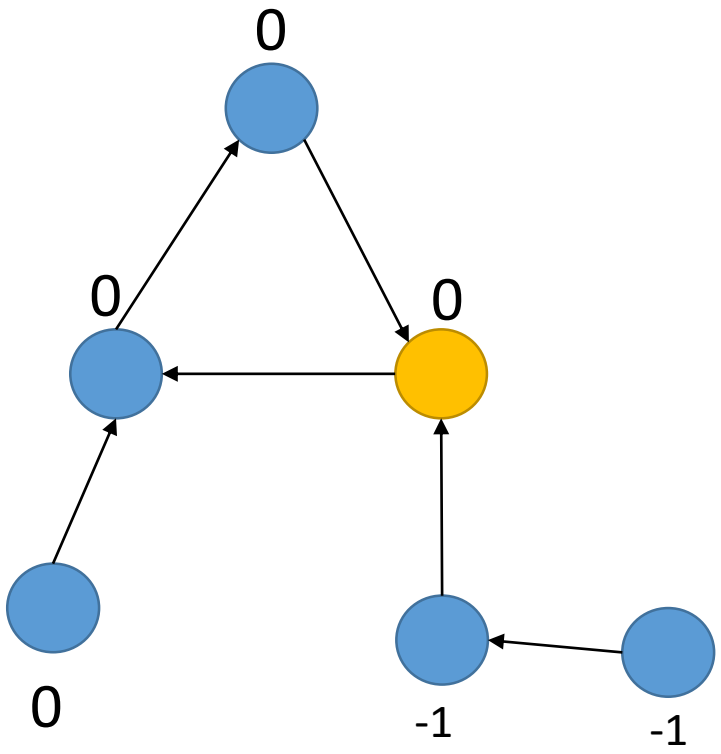
おすすめの実装

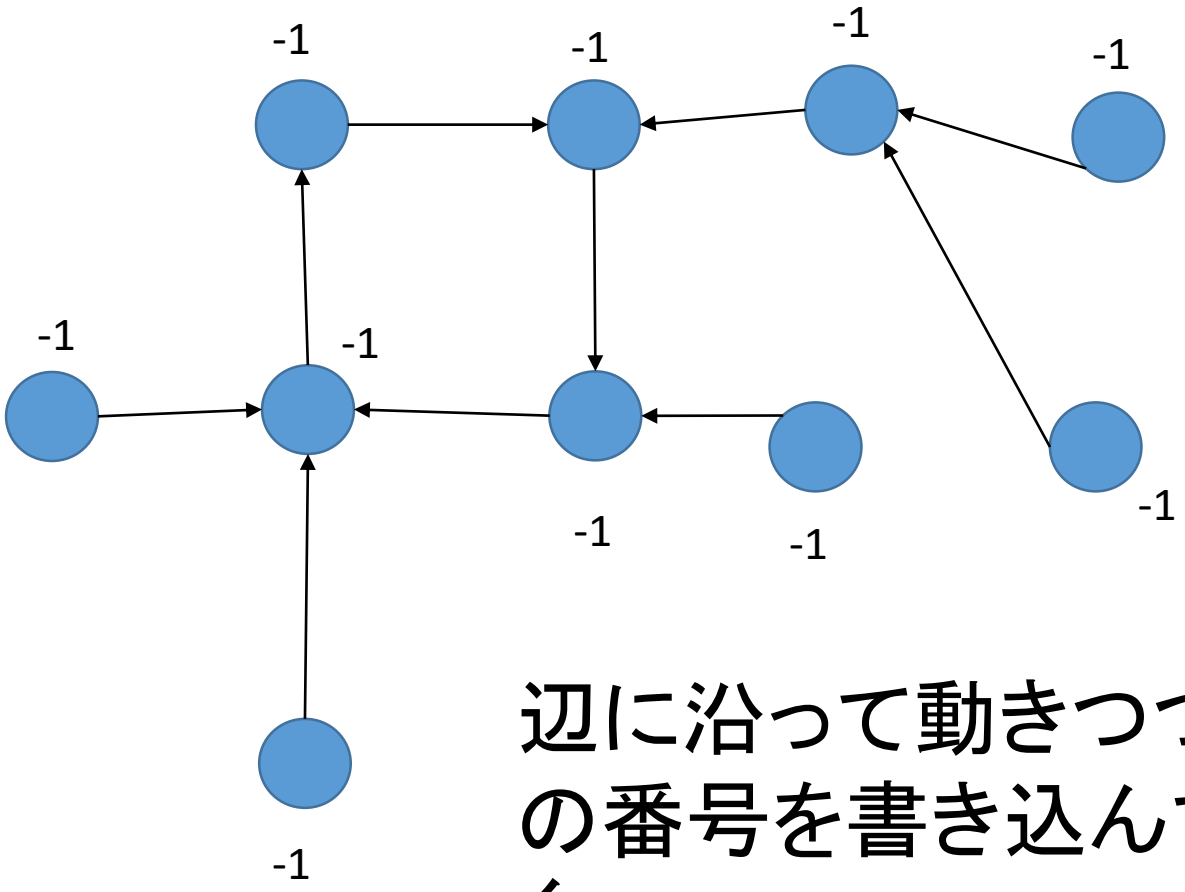




辺に沿って動きつつその番号を書き込んでいく

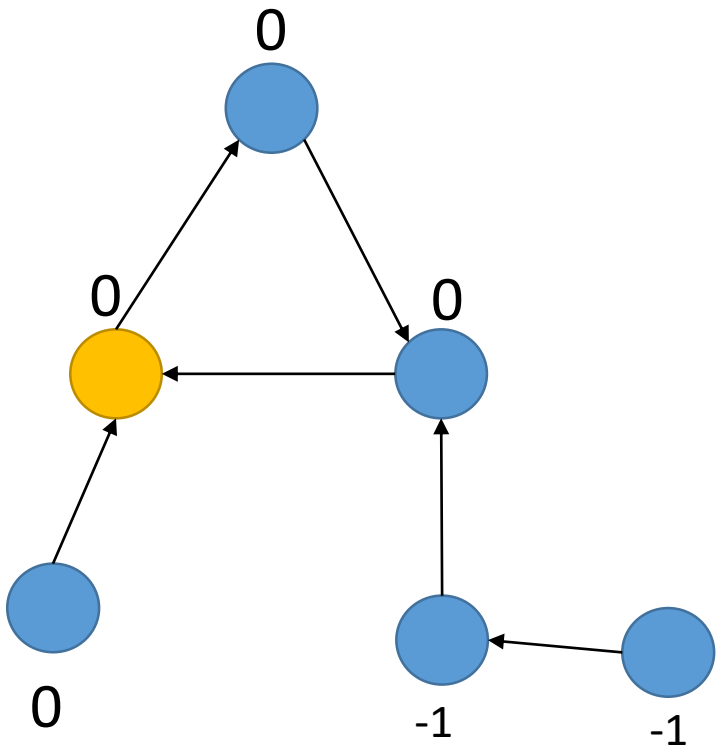
おすすめの実装

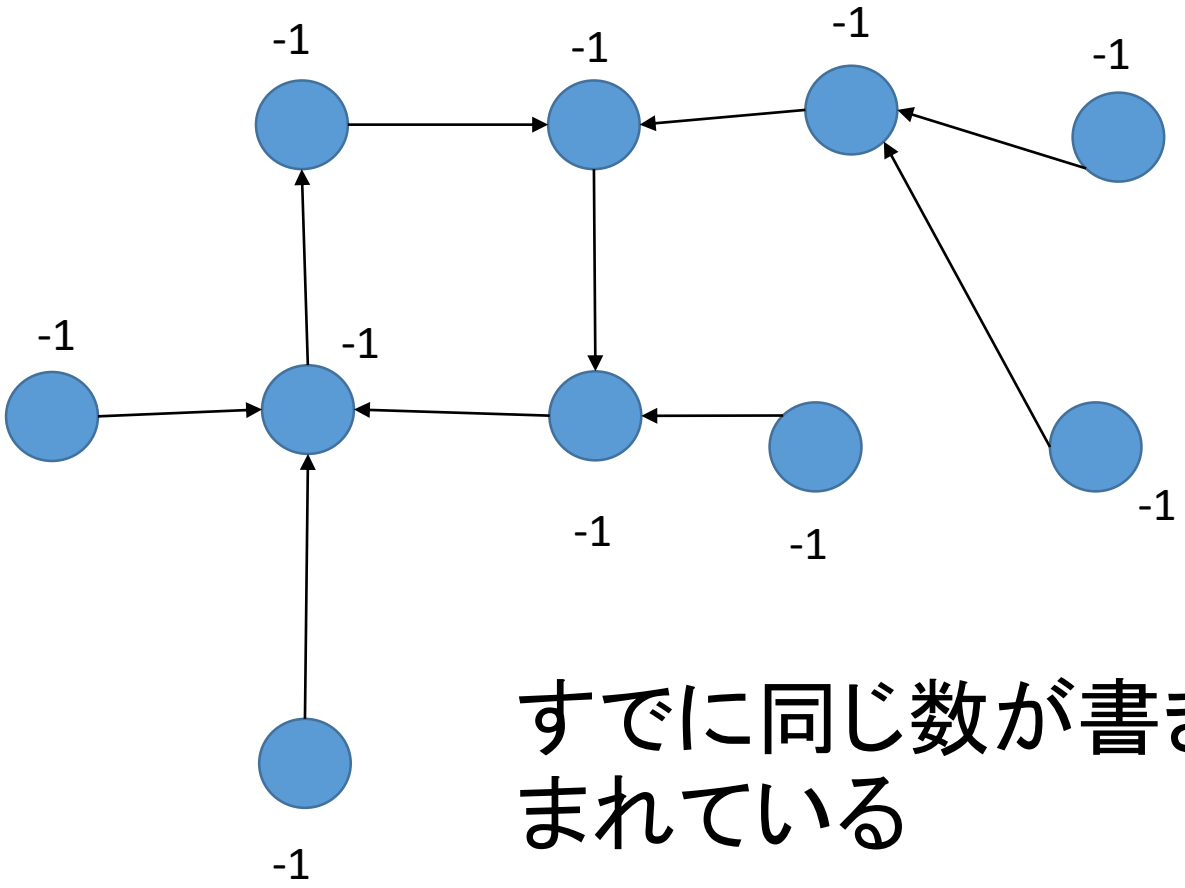




辺に沿って動きつつその番号を書き込んでいく

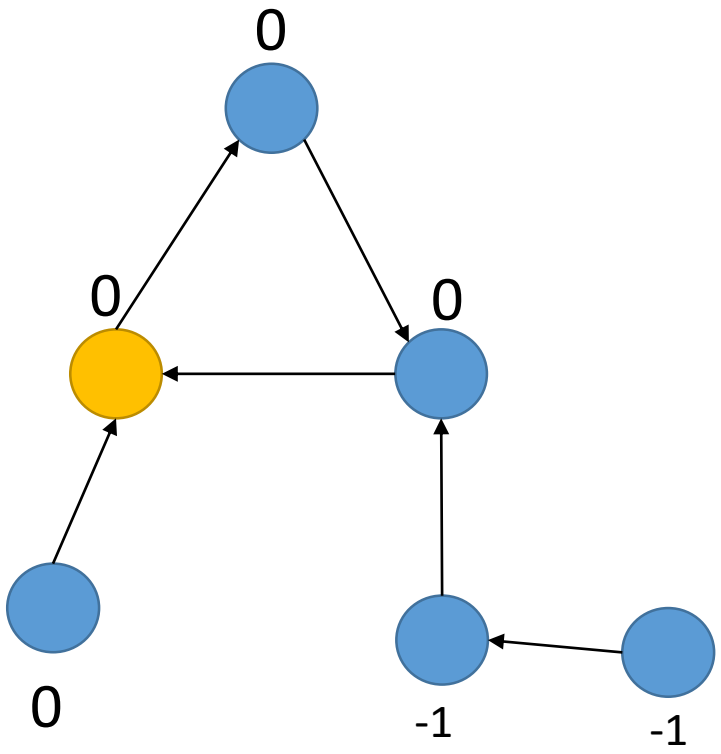
おすすめの実装

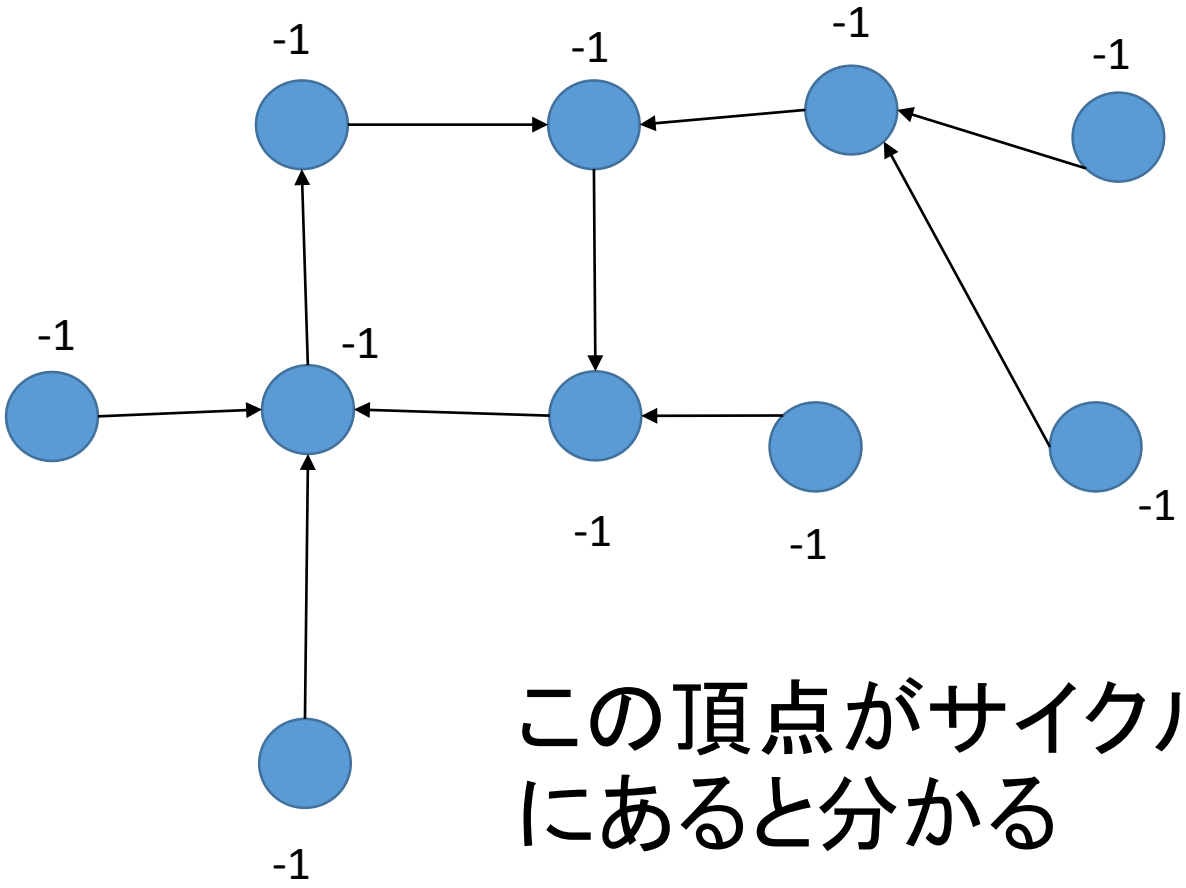




すでに同じ数が書き込まれている

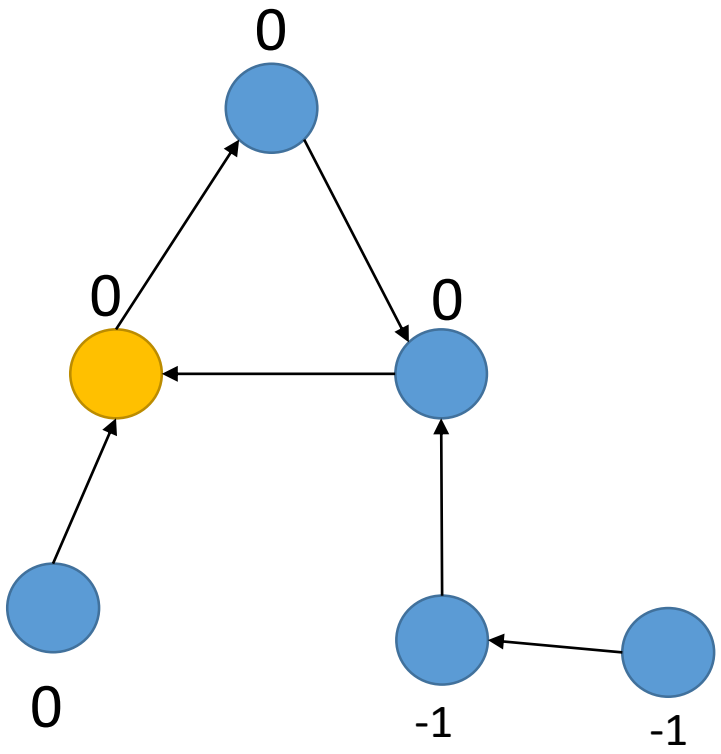
おすすめの実装

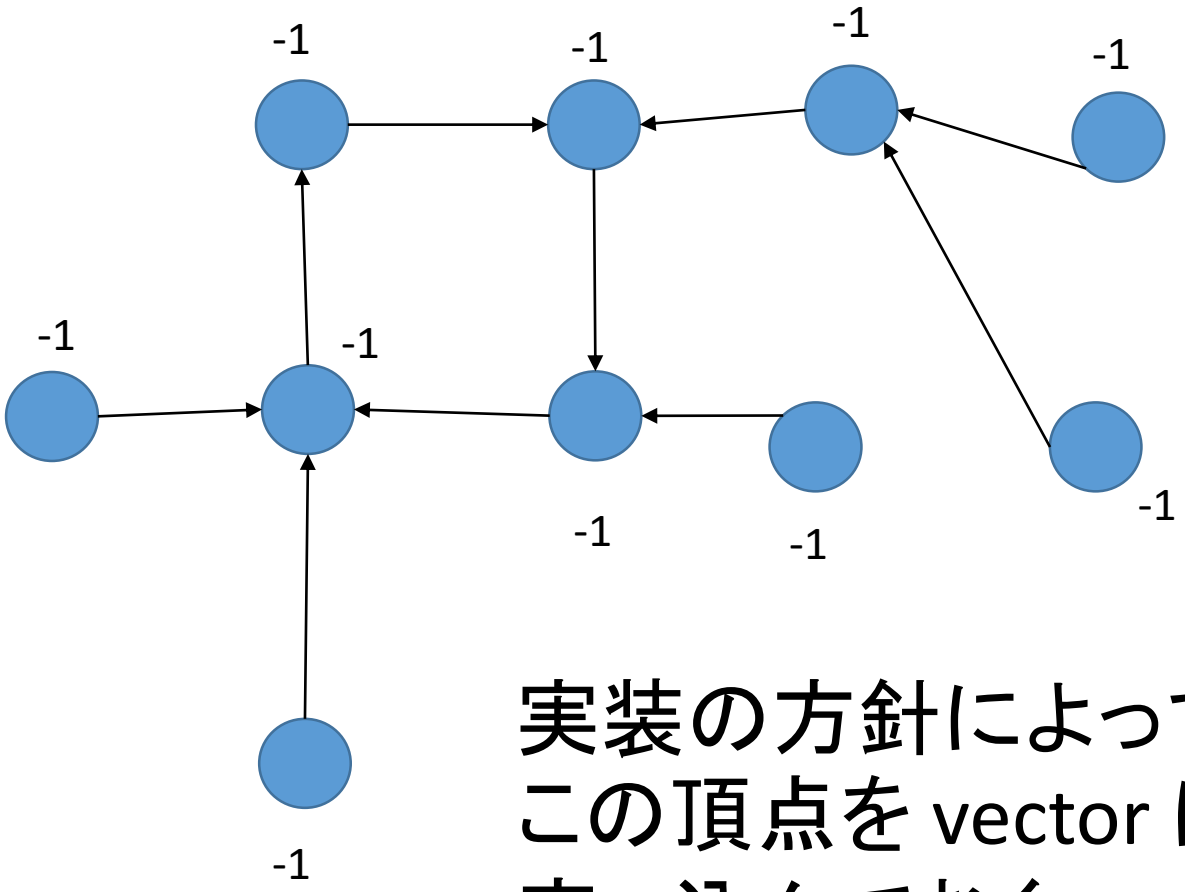




この頂点がサイクル上にあると分かる

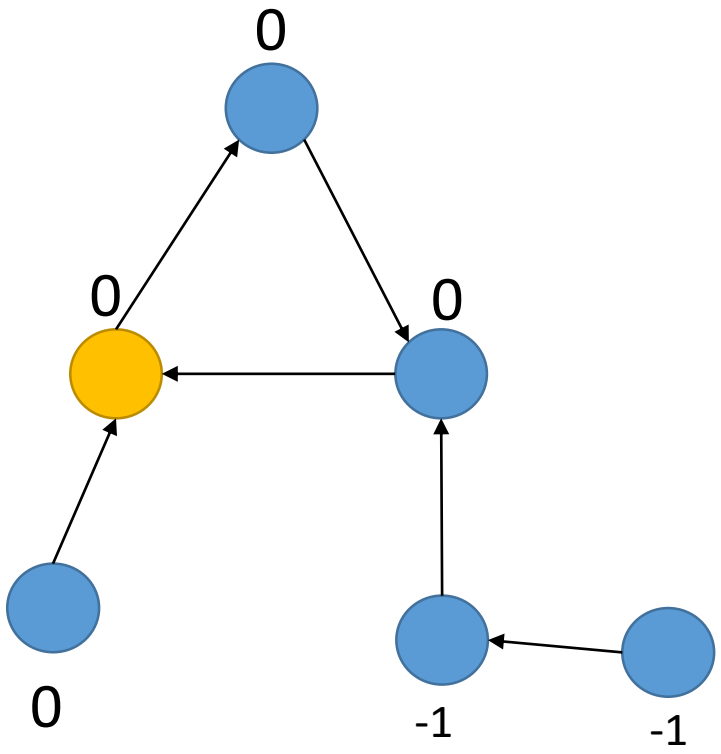
おすすめの実装

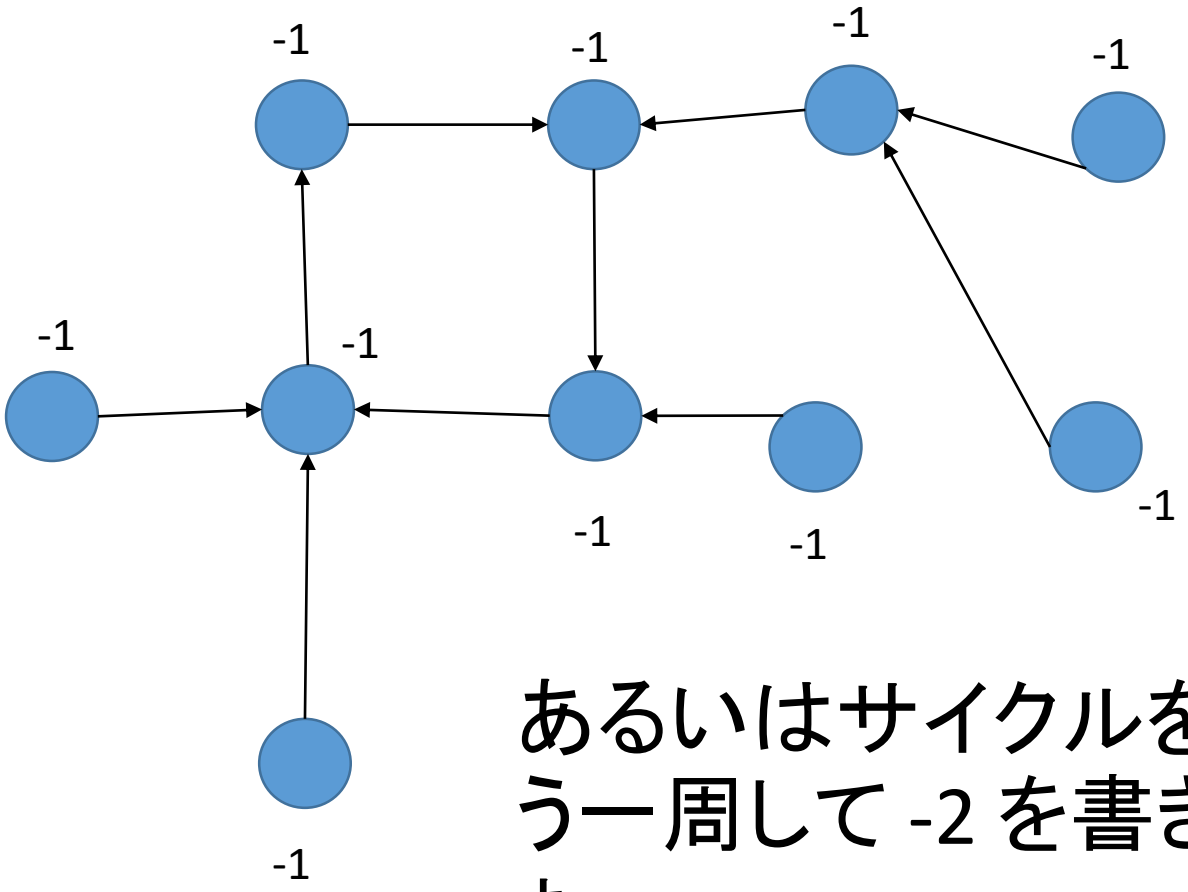




実装の方針によっては
この頂点を vector に
突っ込んでおく

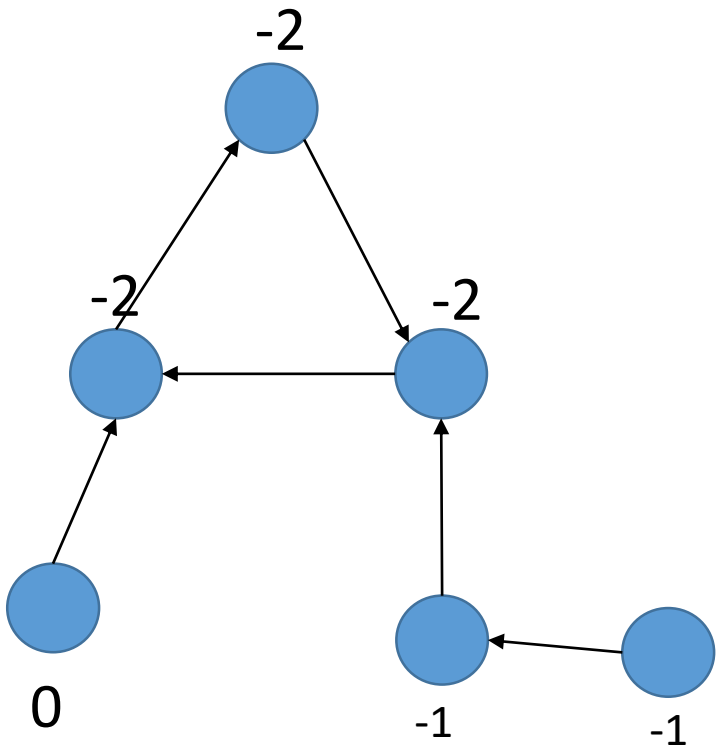
おすすめの実装

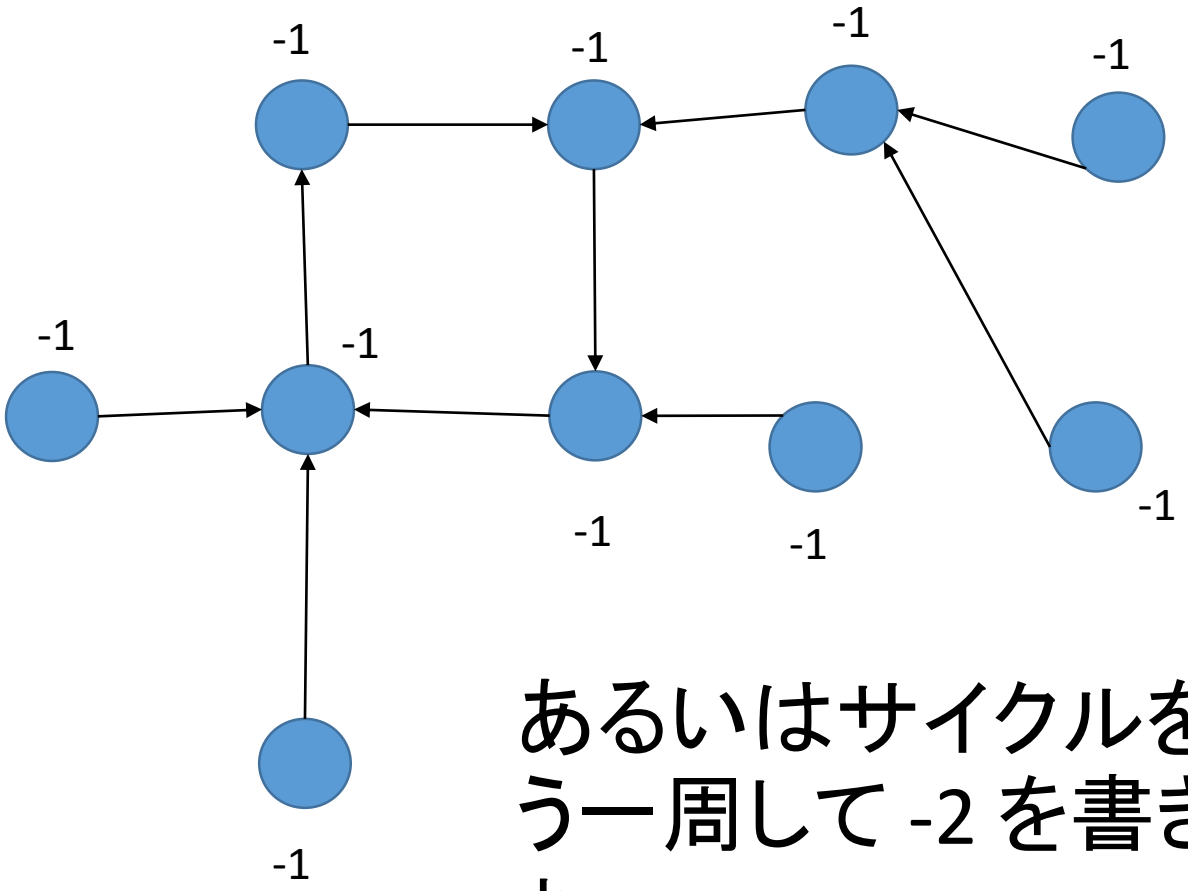




あるいはサイクルをもう一周して -2 を書き込む

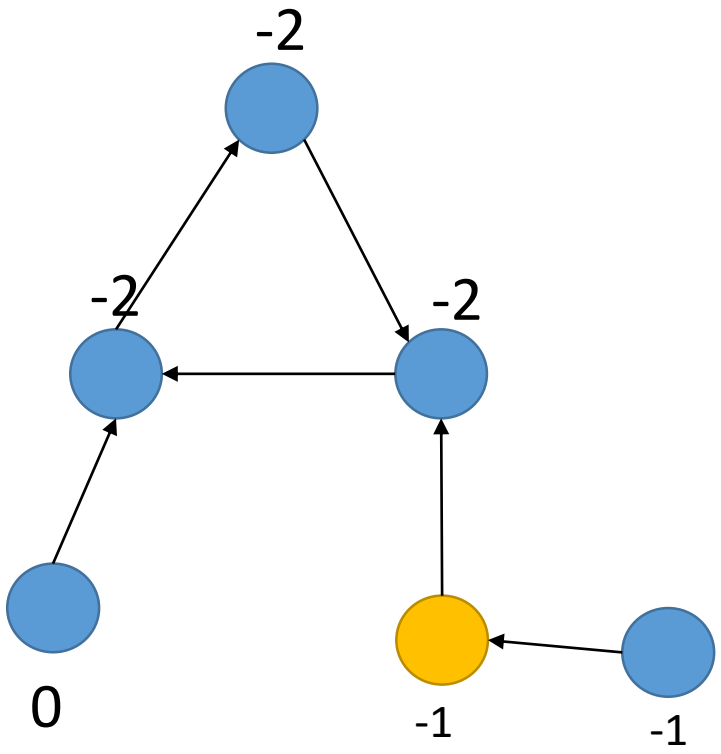
おすすめの実装

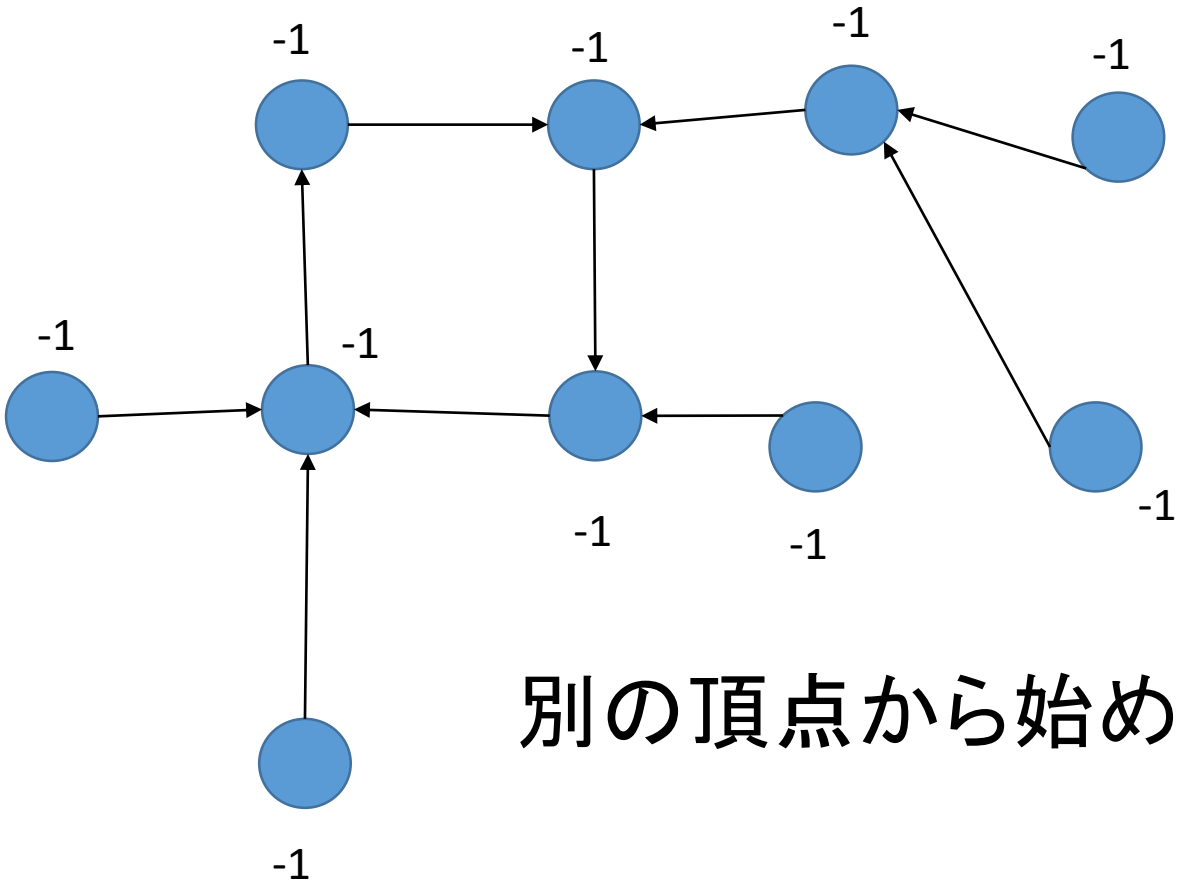




あるいはサイクルをもう一周して -2 を書き込む

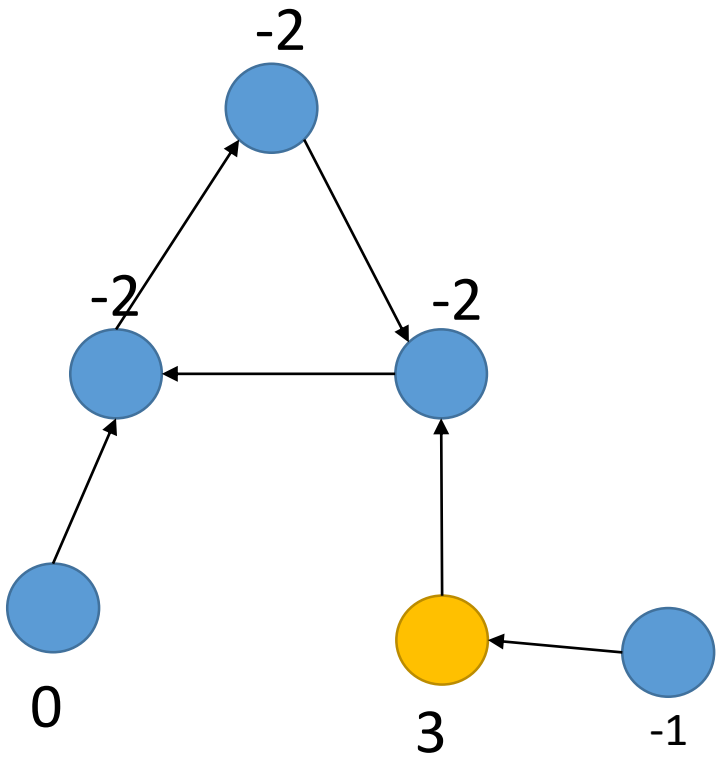
おすすめの実装

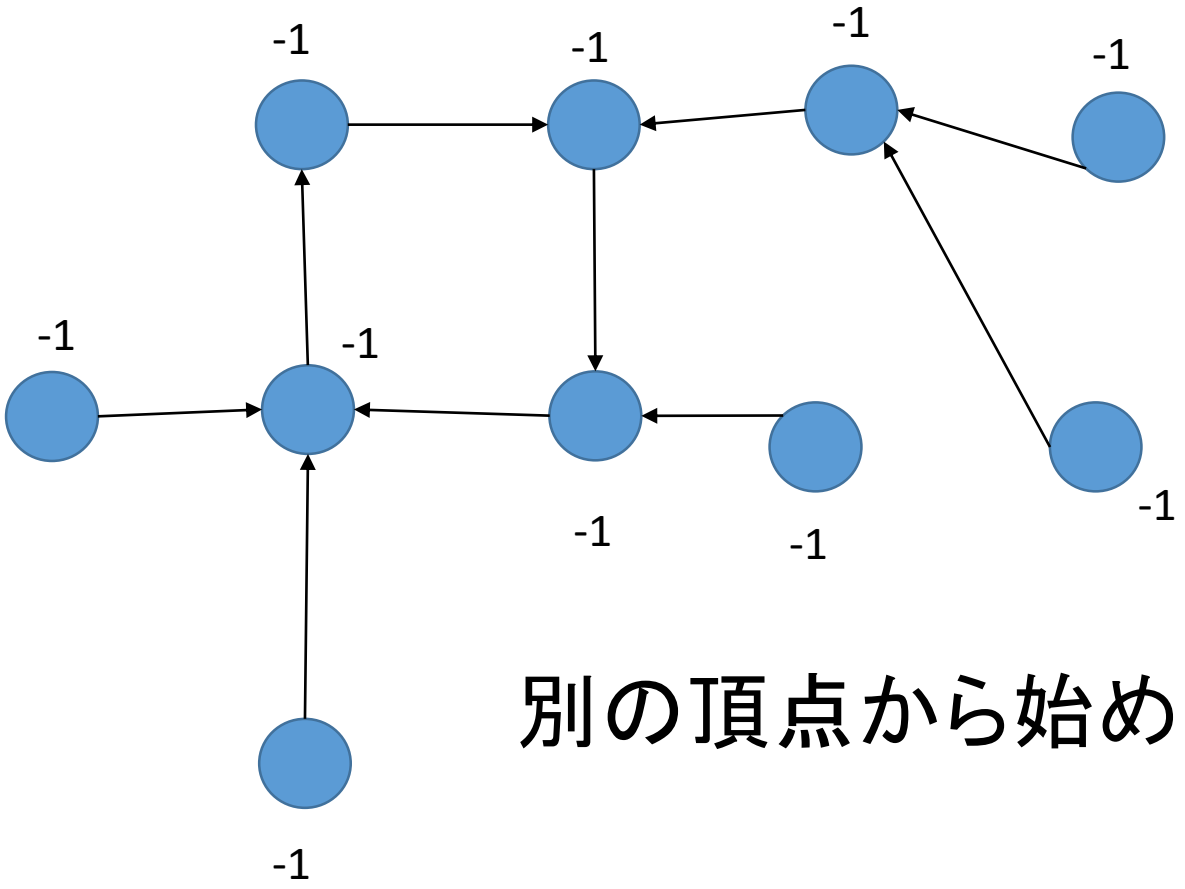




別の頂点から始める

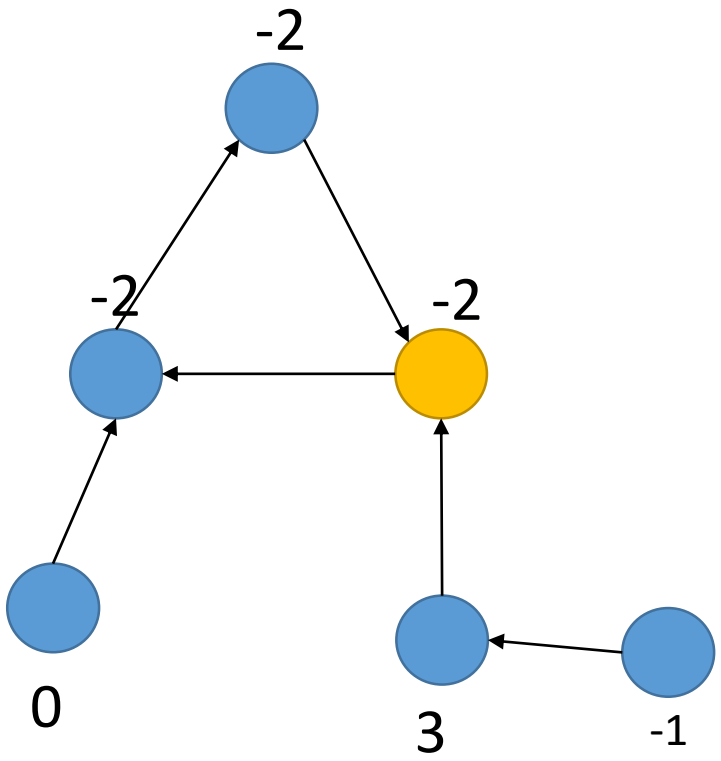
おすすめの実装

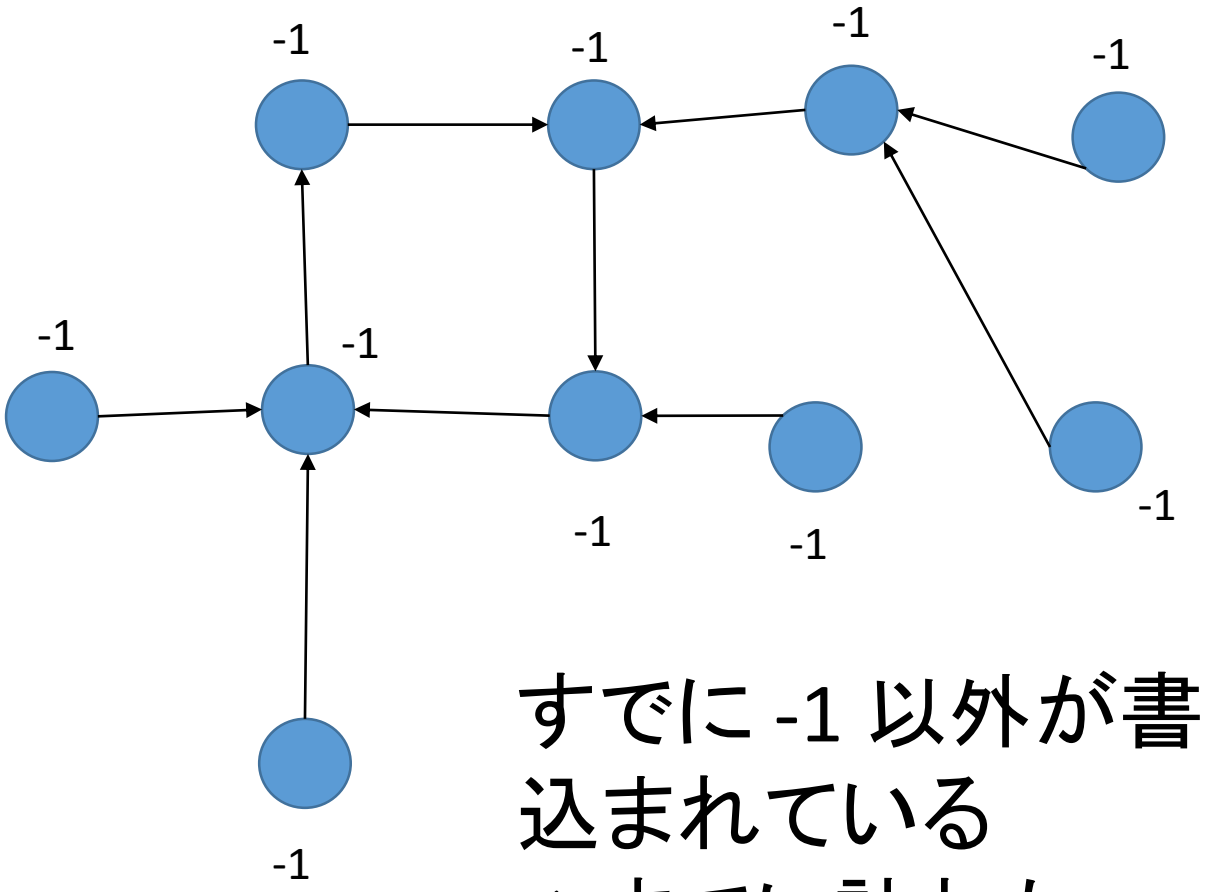




別の頂点から始める

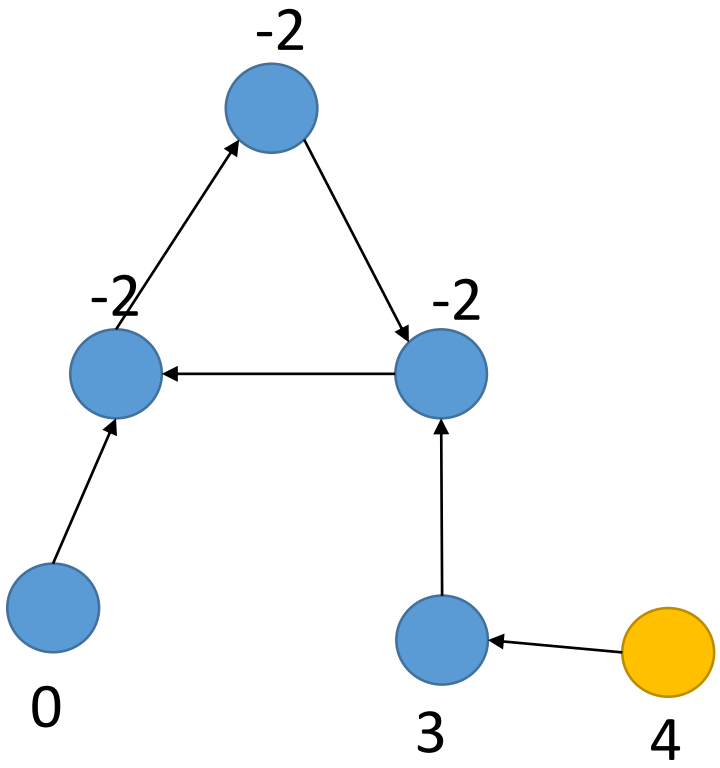
おすすめの実装

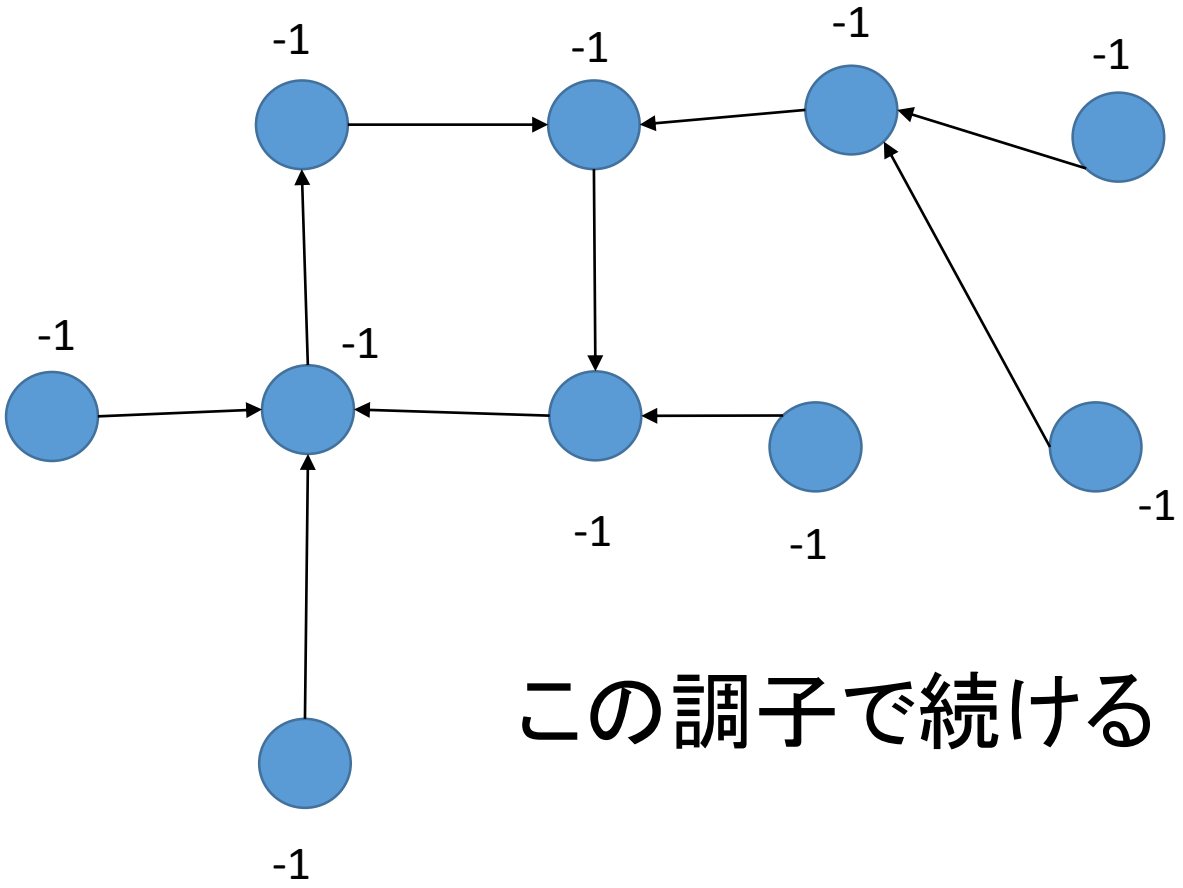




すでに -1 以外が書き込まれている
→すでに訪れた

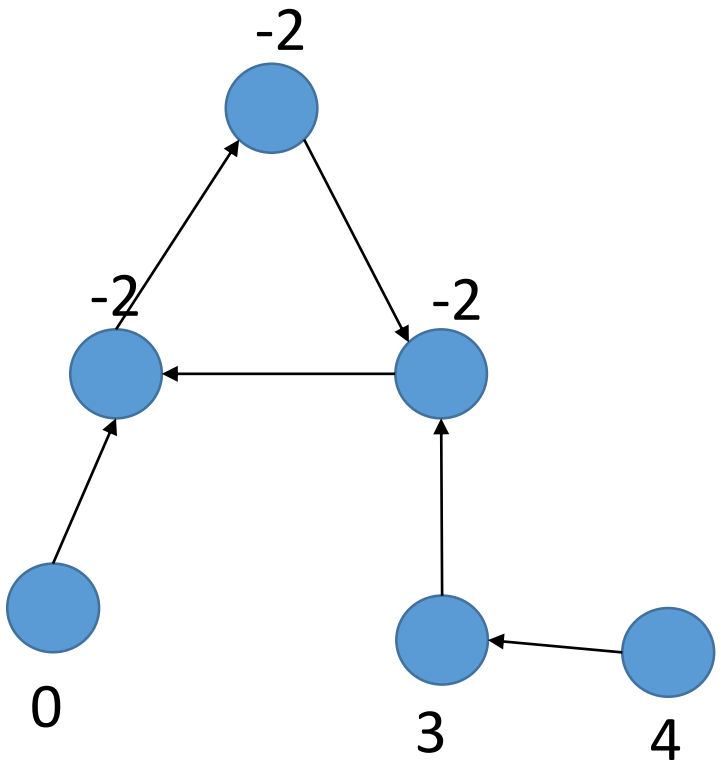
おすすめの実装

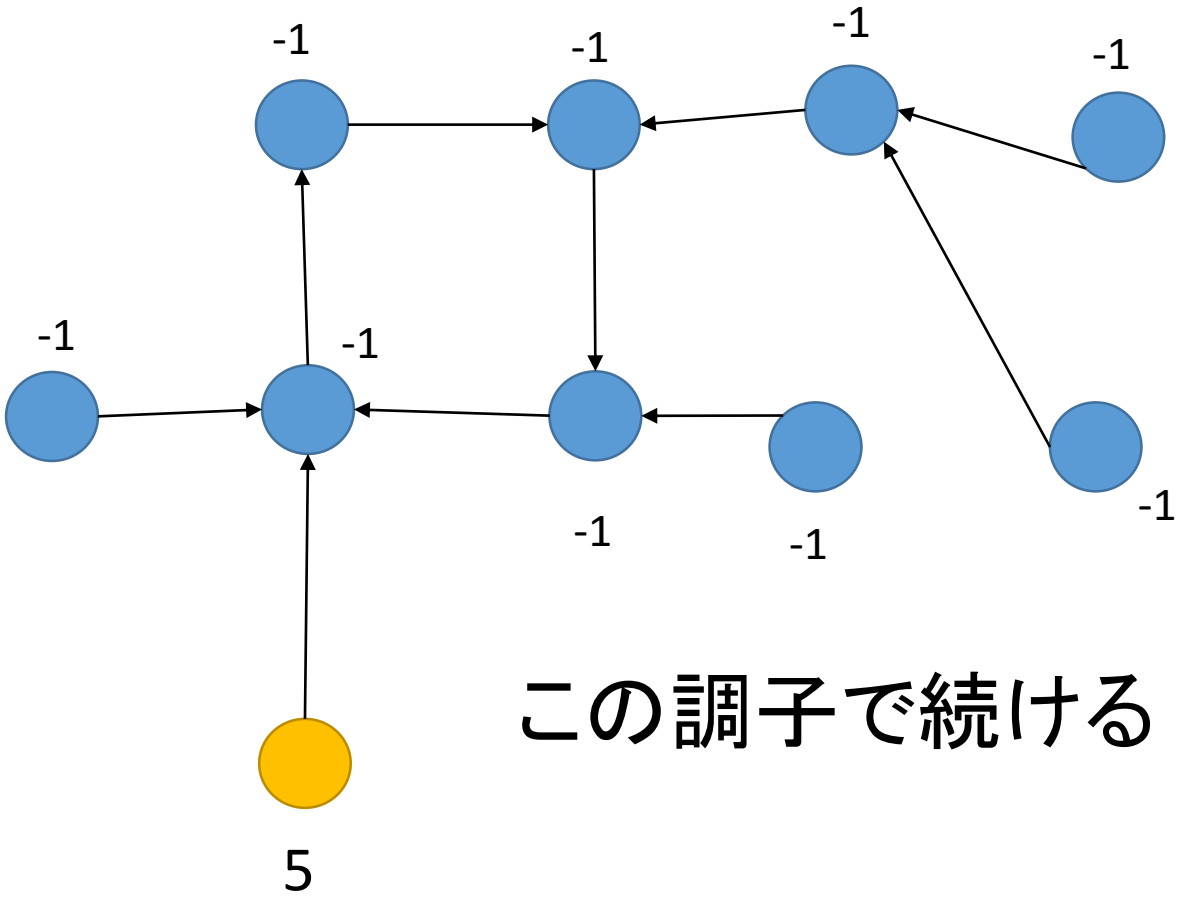




この調子で続ける

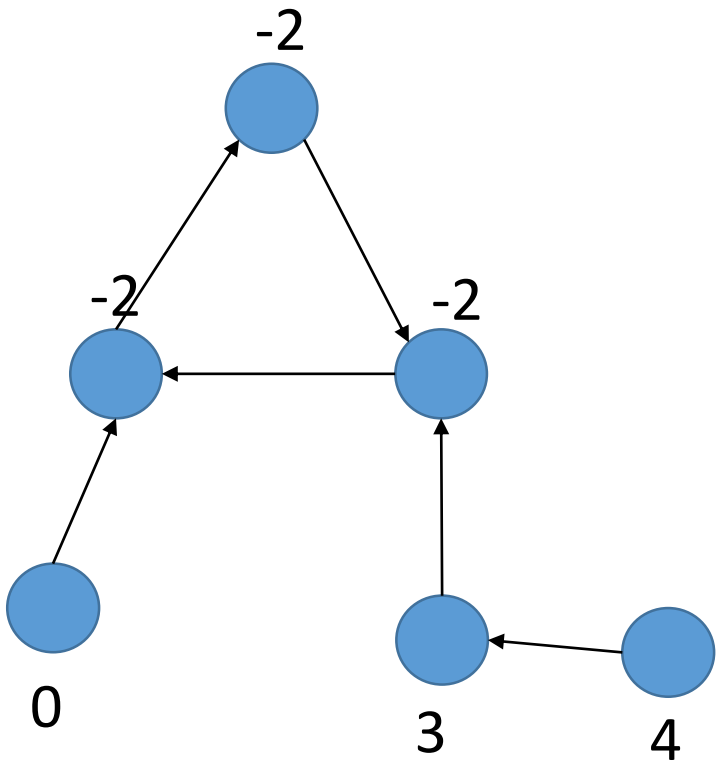
おすすめの実装

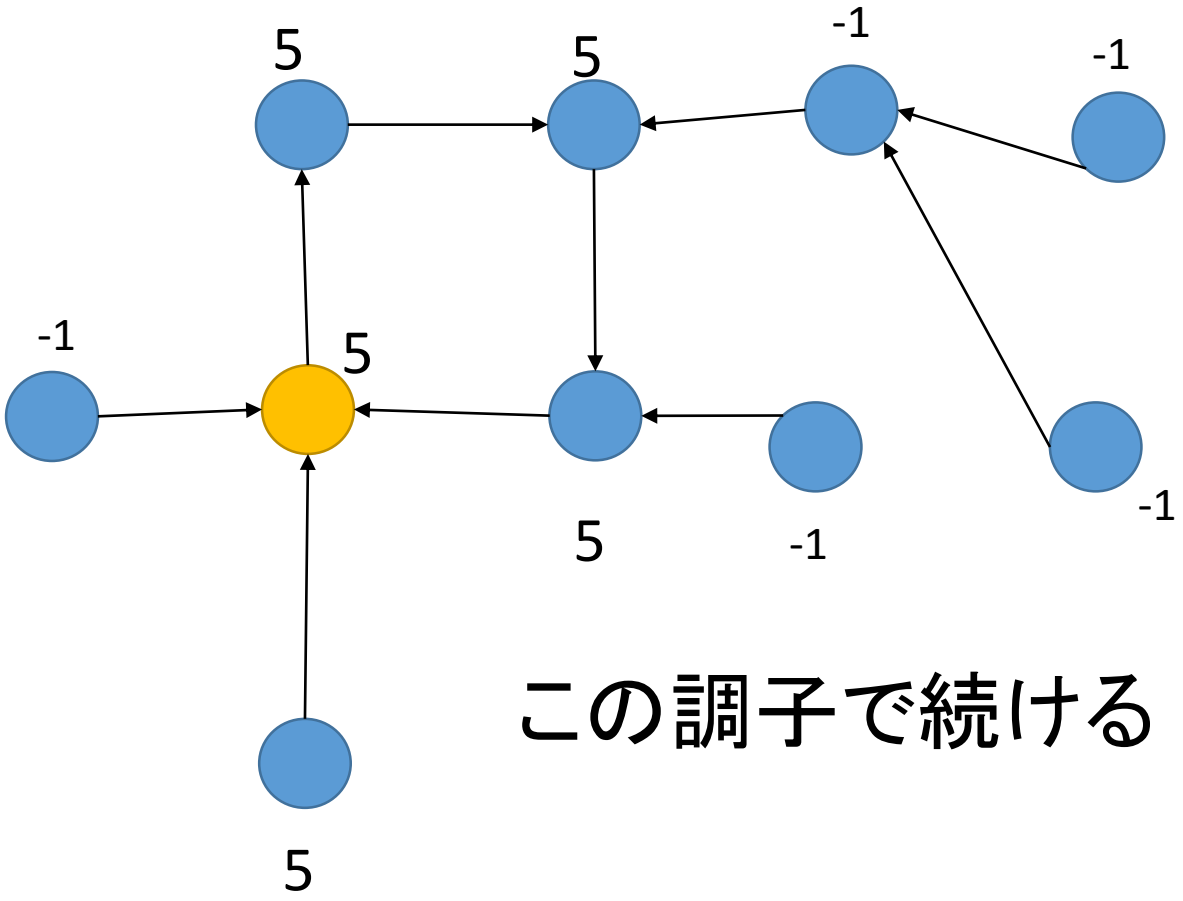




この調子で続ける

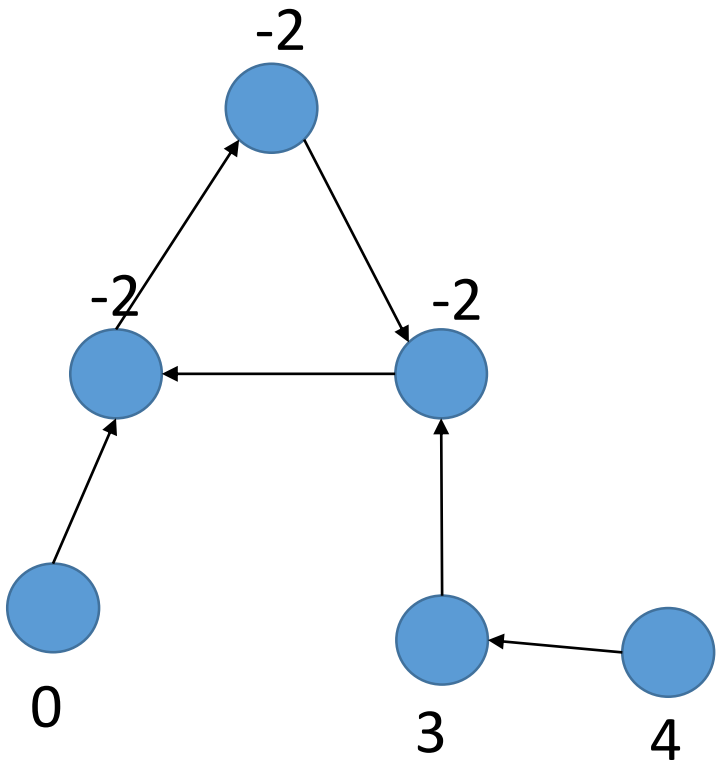
おすすめの実装

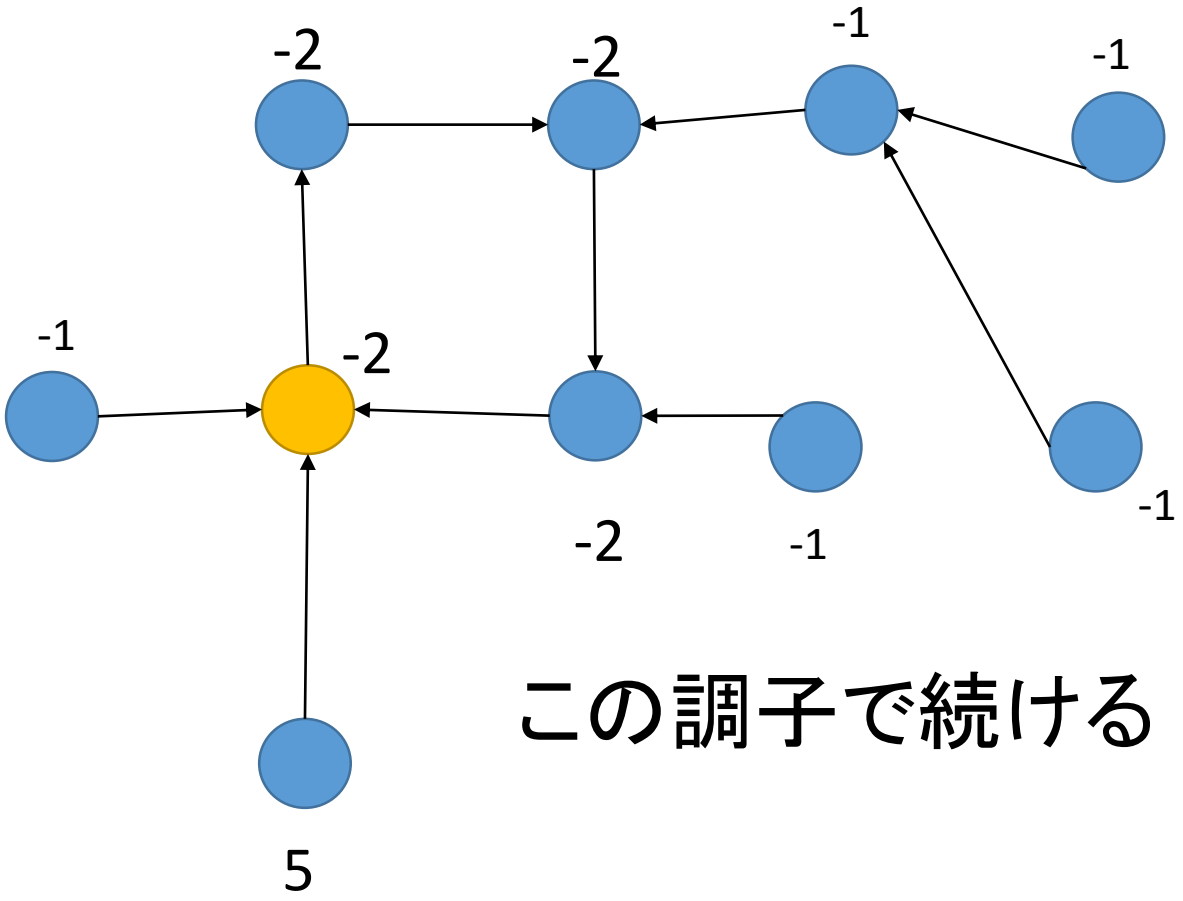




この調子で続ける

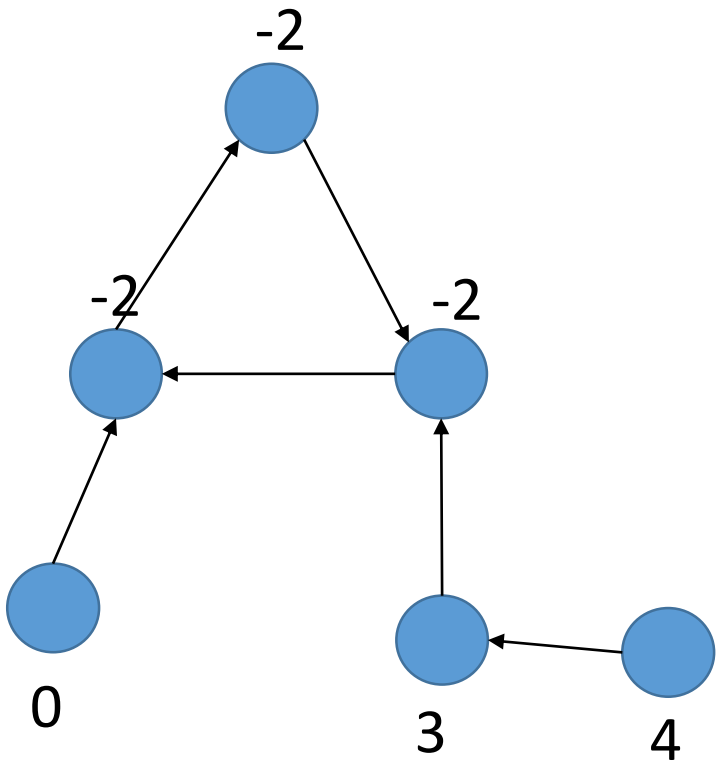
おすすめの実装

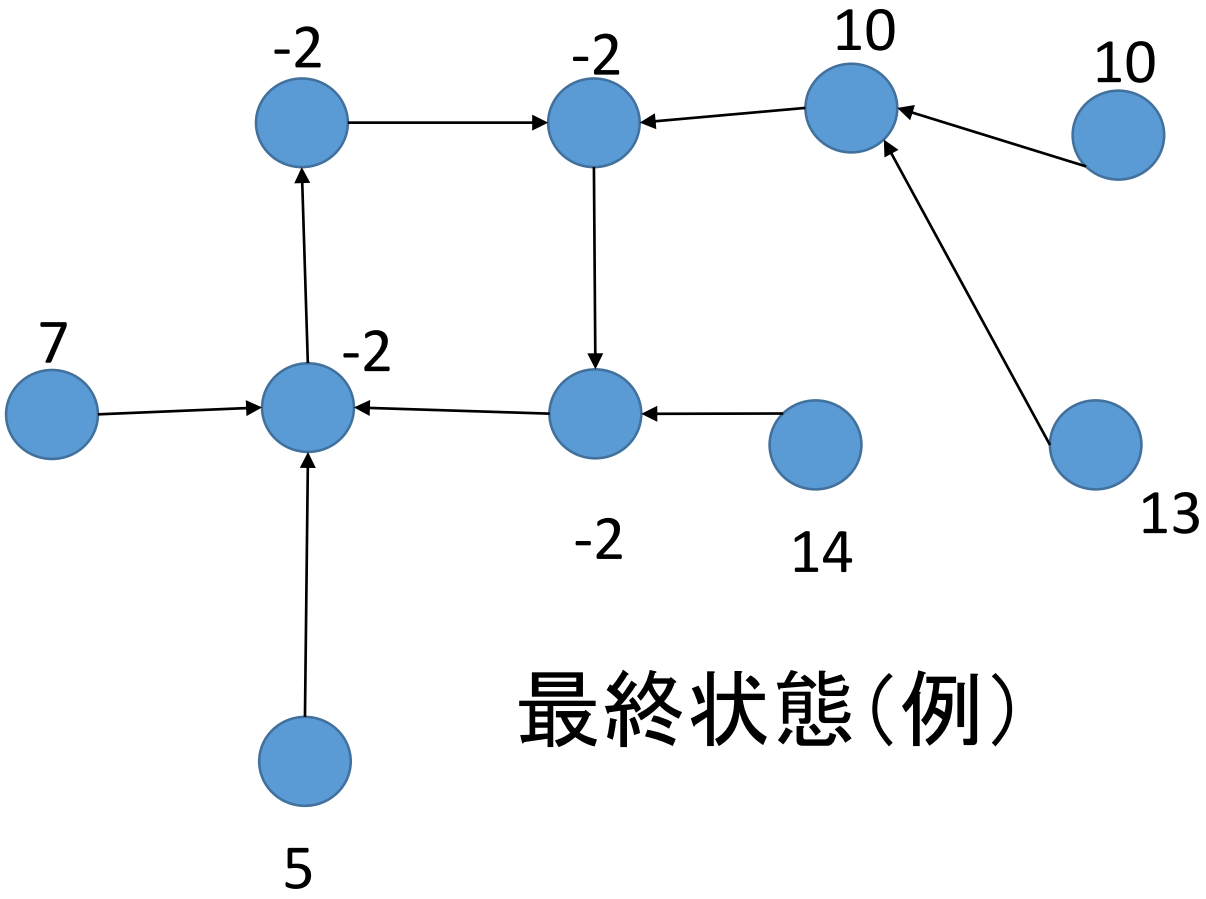




この調子で続ける

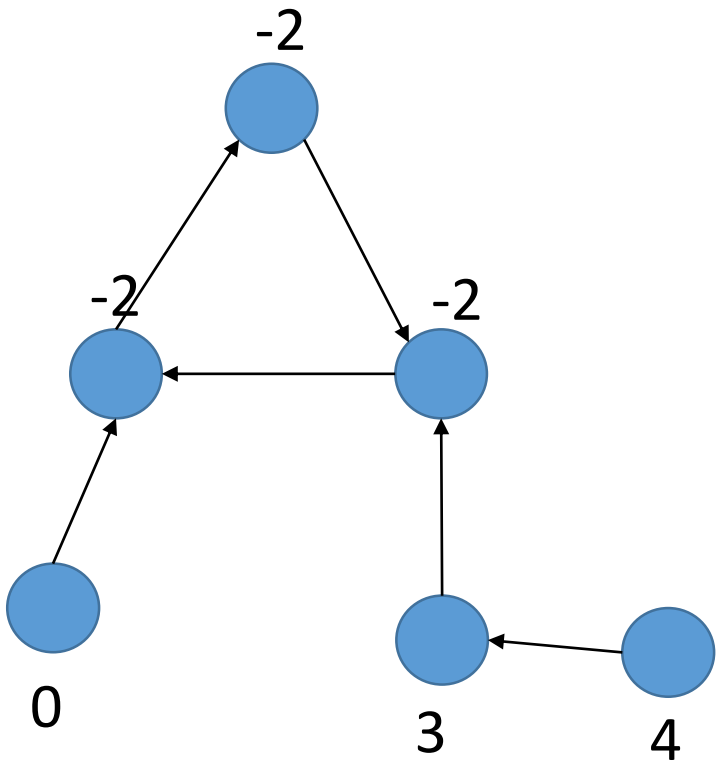
おすすめの実装

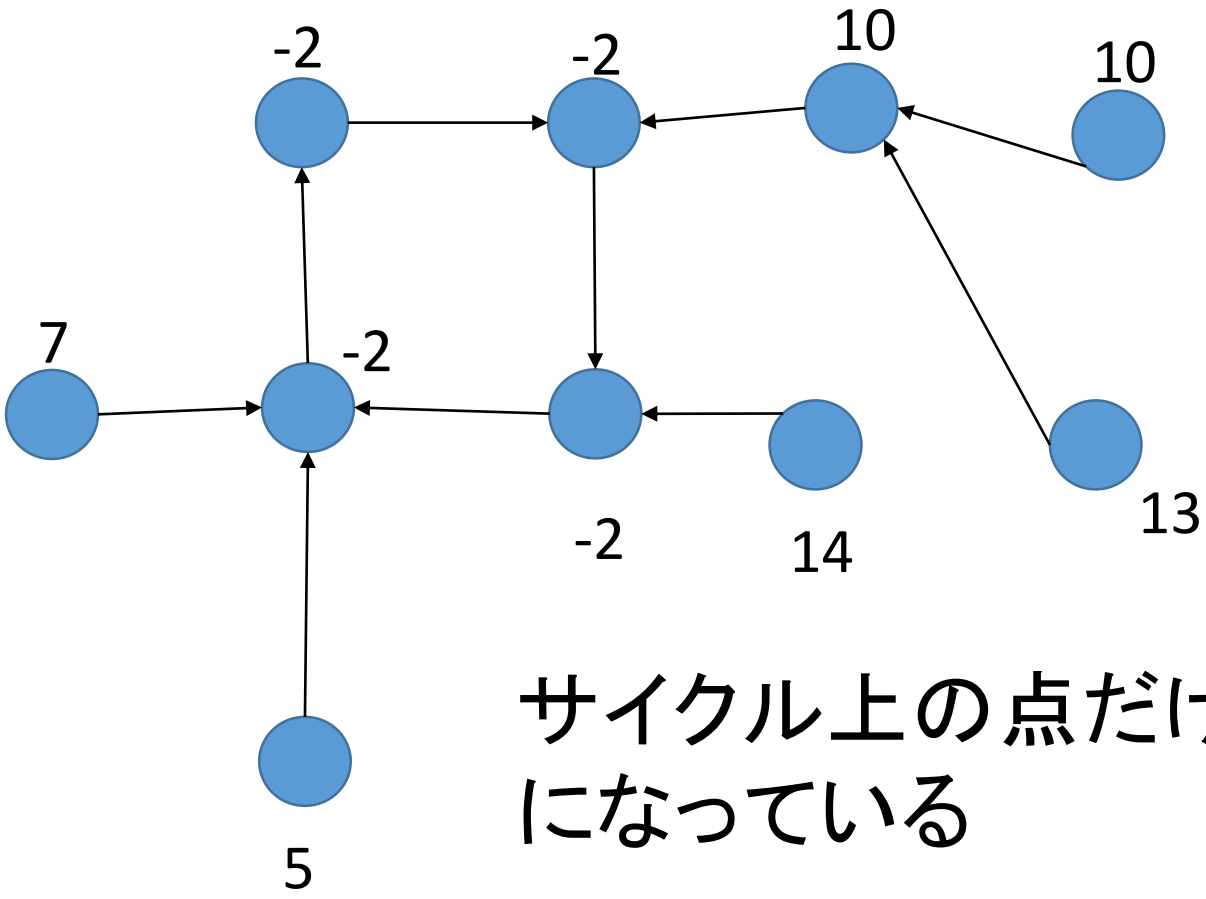




最終状態（例）

おすすめの実装





サイクル上の点だけ -2 になっている

おすすめの実装

得点分布

得点分布