

解説

## Day4 Presents

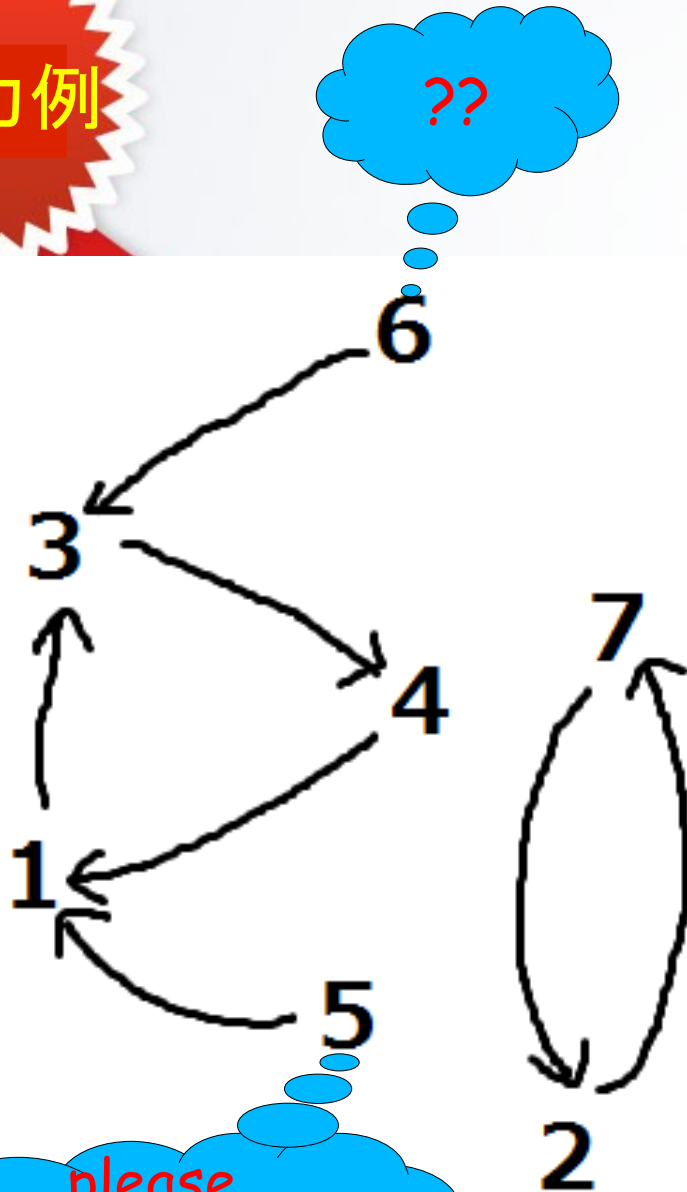


秀 郁未 (tozangezan)

## 概要

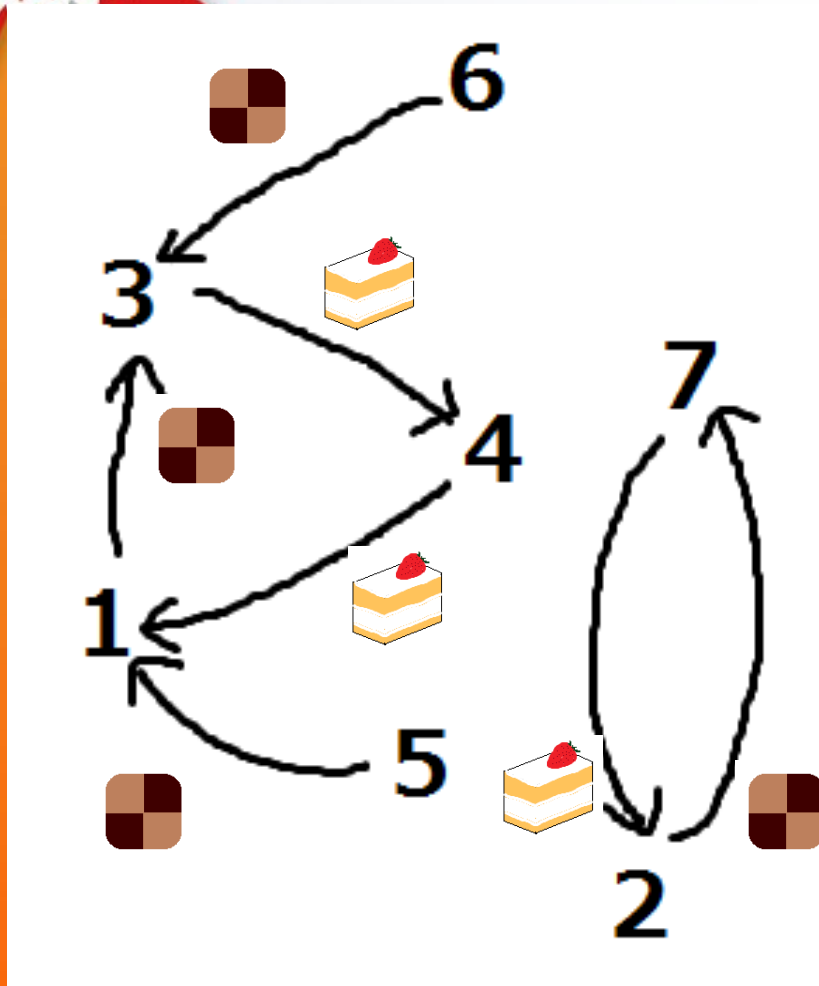
- $N$  人の人がいて、他のある人に 2 種類のお菓子（ケーキかクッキーのどちらか）のうち 1 種類を決めて何個か渡す
- 受け取る人は自分のあげたお菓子と同じお菓子 1 個をもらう、もしくは違うお菓子 1 個をもらうことで得られる嬉しさが決まっている
- それぞれの人がどちらのお菓子を作るか決めることができるので、 $N$  人の嬉しさの合計の最大値を求めよ。
- $N \leq 100,000$

## 入出力例



- こんな感じにお菓子を渡す
- 同じ種類のお菓子をもらうと嬉しい人々  
→ 1,4
- 別の種類のお菓子をもらうと嬉しい人々  
→ 3,7
- どっちでも良い人  
→ 2
- どうせもらえない人  
→ 5,6 (絶望)

## 入出力例



- 1,2,5,6 がクッキー、  
3,4,7 がケーキを作ると、
- 嬉しさの合計は  
 $88+40+90+35+0+0+4=257$   
で最大



素朴  
な  
解法

- それぞれの人がケーキ、クッキーどちらかを渡すか決めてシミュレーション
- 決め方が  $O(2^N)$ 、それぞれについてシミュレーションは  $O(N)$ 。計  $O(2^N N)$  → 10 点

## 考察

- 見た感じグラフの問題
- どんな感じの形になるのだろうか??
- 各頂点からは辺が必ず1本だけ出ている  
( $\sim \ominus \sim$ )。o(待てよ、こんな感じのグラフは過去にも出題されてたような…)

| JOI 合宿 過去問 |  $\sqsubset (\cup \setminus ) \sqsupset$  三

## 報告 (Report)

情報オリンピック日本委員会は報告・連絡・相談が非常に徹底された組織である。委員会には  $N$  人が所属し、すべての人に対しその報告を受ける人(「報告先」と呼ぶ)が 1 人ずつ定まっている。

今、情報オリンピック日本委員会の中で、あるプロジェクトを立ち上げることとなった。このプロジェクトは、仕事 1, 仕事 2, ..., 仕事  $N$  という  $N$  個の仕事に分けられており、 $N$  人には一つずつ異なる担当が割り当てられている。 $N$  個の仕事は仕事 1 から仕事  $N$  まで番号順に処理される。ある仕事の担当者がその仕事を終えたとき、その人は自分の「報告先」に対して「作業報告」を行う。「作業報告」を受けた人は、それと同一の「作業報告」を自分の「報告先」に対して行う。ただし、同じ「作業報告」を既に自分の「報告先」にしたことがある場合は、同じ仕事に対する「作業報告」を再びすることはない。これが繰り返されることで、委員会中の何人かが「作業報告」を受けることとなる。次の仕事は、前の仕事の「作業報告」が全て行われてから処理される。

### 課題

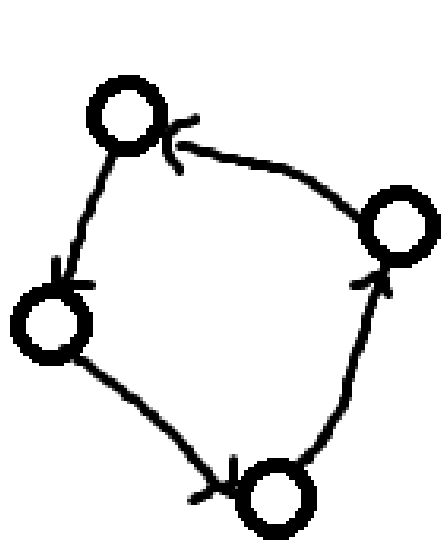
入力として各仕事の担当者の「報告先」が与えられたとき、それぞれの人が自分の仕事に取り組む段階において、受けた「作業報告」の種類数を出力するプログラムを作成せよ。

### 制限

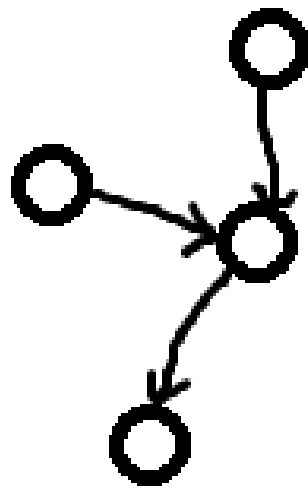
$2 \leq N \leq 100\,000$  委員会に所属する人数

## 思考

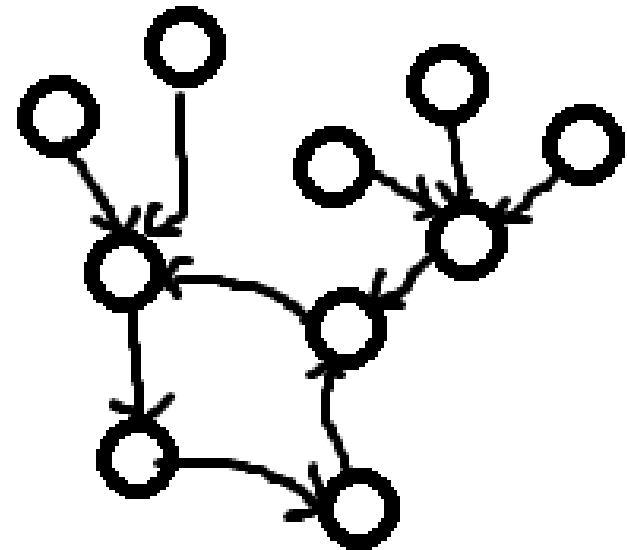
- どんな感じの形になるかいろいろ作って確かめてみる



こんなのや



こんなの



こんなのも!!

- 輪におまけの木がついたような形??
- これらが複数存在することも

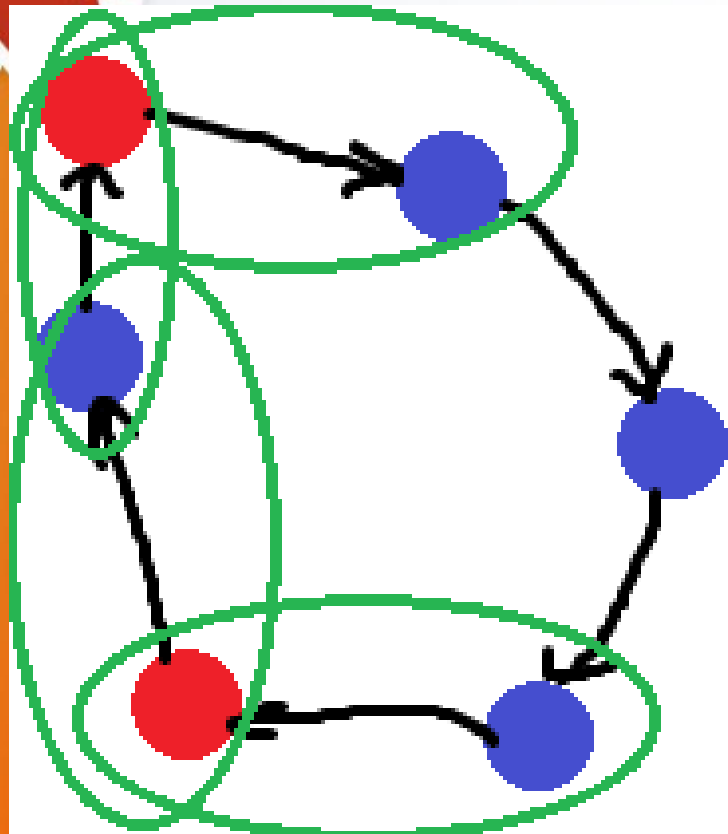




## 満点解法

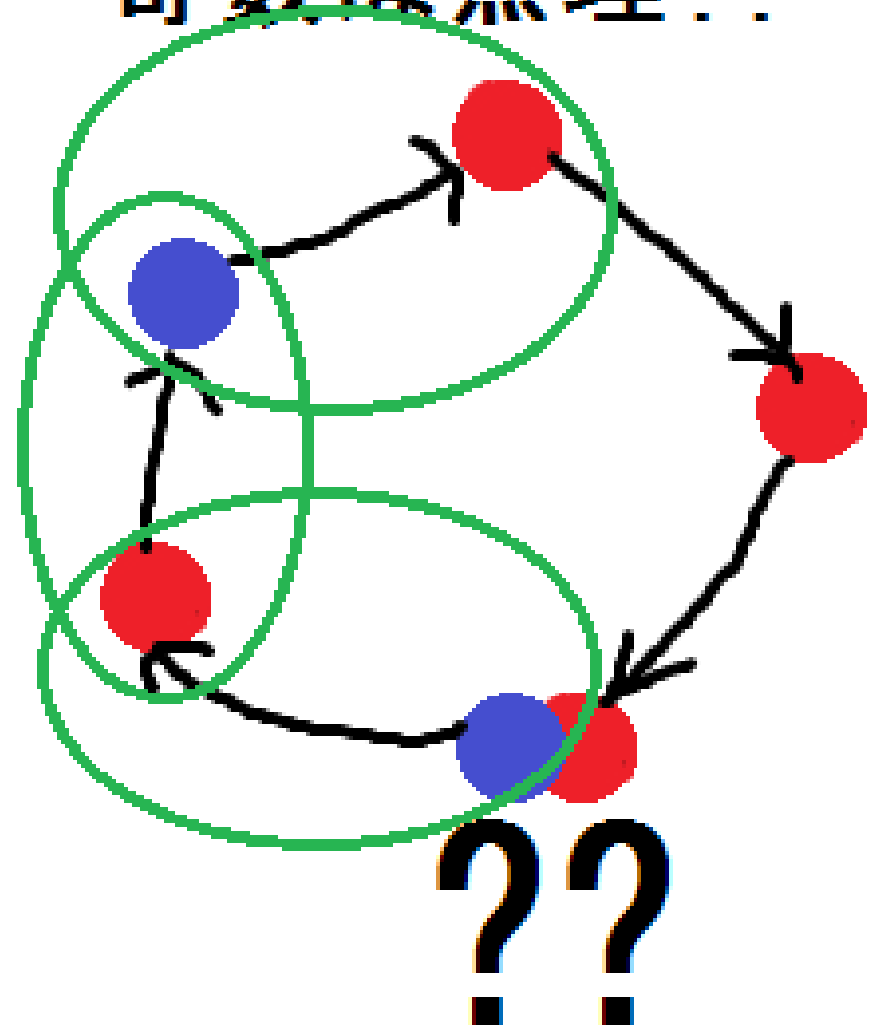
- まず、サイクル部分について考えることにする。
- 各人がクッキー、ケーキどちらかを渡すか、ということよりもらう物が自分があげた物と同じか違うか、という考え方のほうが考えやすい。
- サイクル内では「自分のあげた物と違う物を（サイクル内にいる人に）もらった」という人は偶数人でなければならない。

図説



この例だと4回  
変わっている

奇数は無理!!



## 満点 解法

- $i$  番目までで  $j$  人「違うお菓子をもらう」ときの嬉しさの合計の最大値を  $dp[i][j]$  とすると  
→  $O(N^2)$
- そもそも DP しなくても Greedy で十分！
- $\{(\text{違う嬉しさ}) - (\text{同じ嬉しさ})\} * (\text{もらう個数})$  をソートして、大きい順に偶数個取ったときの合計の最大値を取ればよい
- すこし考えればソートしないで  $O(N)$  でも出来る。(偶奇で DP)



## 満点解法

- サイクルでない部分→根（サイクルに含まれるところ）から順に Greedy するだけ
- ここではクッキー、ケーキのことを全く考えず「同じ」「違う」のみを考えるべき
- 「違うものがほしい」人には違うものを渡し、「同じものがほしい」人には同じものを渡す。複数人で競合しないのでこれで OK。



## まとめ

- 全探索だと  $O(2^N N)$  で 10 点
- サイクルの中で最大値を求めるのに DP を使うと  $O(N^2)$  で 30 点
- Greedy なら  $O(N \log N)$  または  $O(N)$  で 100 点
- ということで、



## 実装

- この問題の実装でいちばんやりにくいところは、サイクルあたりの実装
- いろいろやり方あると思うので説明していきます



## 実装例

### 1

# 頂点にいくつか属性をつける

- まだ訪れていない点、訪れたがサイクルの内部の点か分からない点、サイクルであると分かった点、サイクルでないと分かった点などの4つ程度に種類を分ける。
- 最初各頂点から辺をたどりたいが、工夫せず全部たどると  $O(N^2)$  となるので絶望
- 一度たどった点はたどらずにループを抜ける（メモ化風）なら  $O(N)$



実装例

1

## 頂点にいくつか属性をつける

- サイクルが複数ある場合というものがあるので、すこし実装に戸惑う。
- いろいろ頂点に種類つけるのが好きな人はこういう実装よくしてそう。





実装例  
2

## 強連結成分分解

- 概要は蟻本や Spaghetti Source で。
- この問題では1つのサイクルが1つの強連結成分で、それ以外の頂点は一つ一つが別々の強連結成分。それぞれのサイズを見ればサイクルの点か判別できそう。
- 実装無駄に長いが、慣れてたら一瞬で書ける、という人もいるかもしれない。



実装例

3

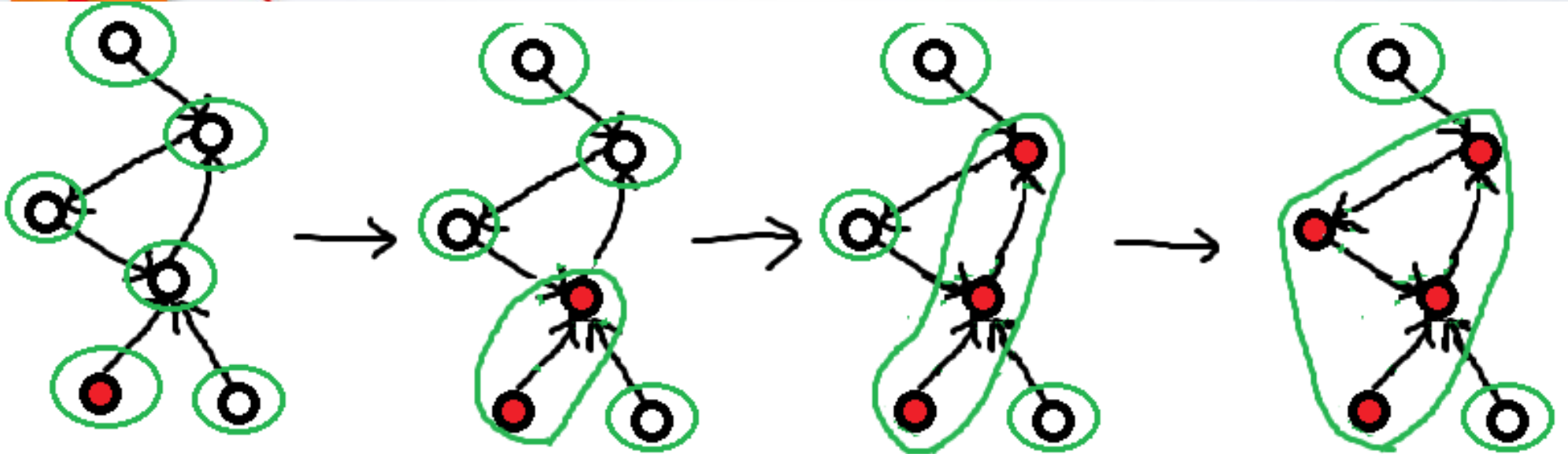
# Union-Find

- 概要は蟻本や Spaghetti Source で。
- 辺をたどるごとに両方の端点が含まれる集合を統合する
- その前から端点と同じ集合に属していたらサイクルがある
- 結局辺を一つ一つたどるので実装があんまり楽ではなさそう。

# 実装例

## 3

# Union-Find



- 同じ集合になったらそのときに見ている頂点からいける点は全部サイクルの中



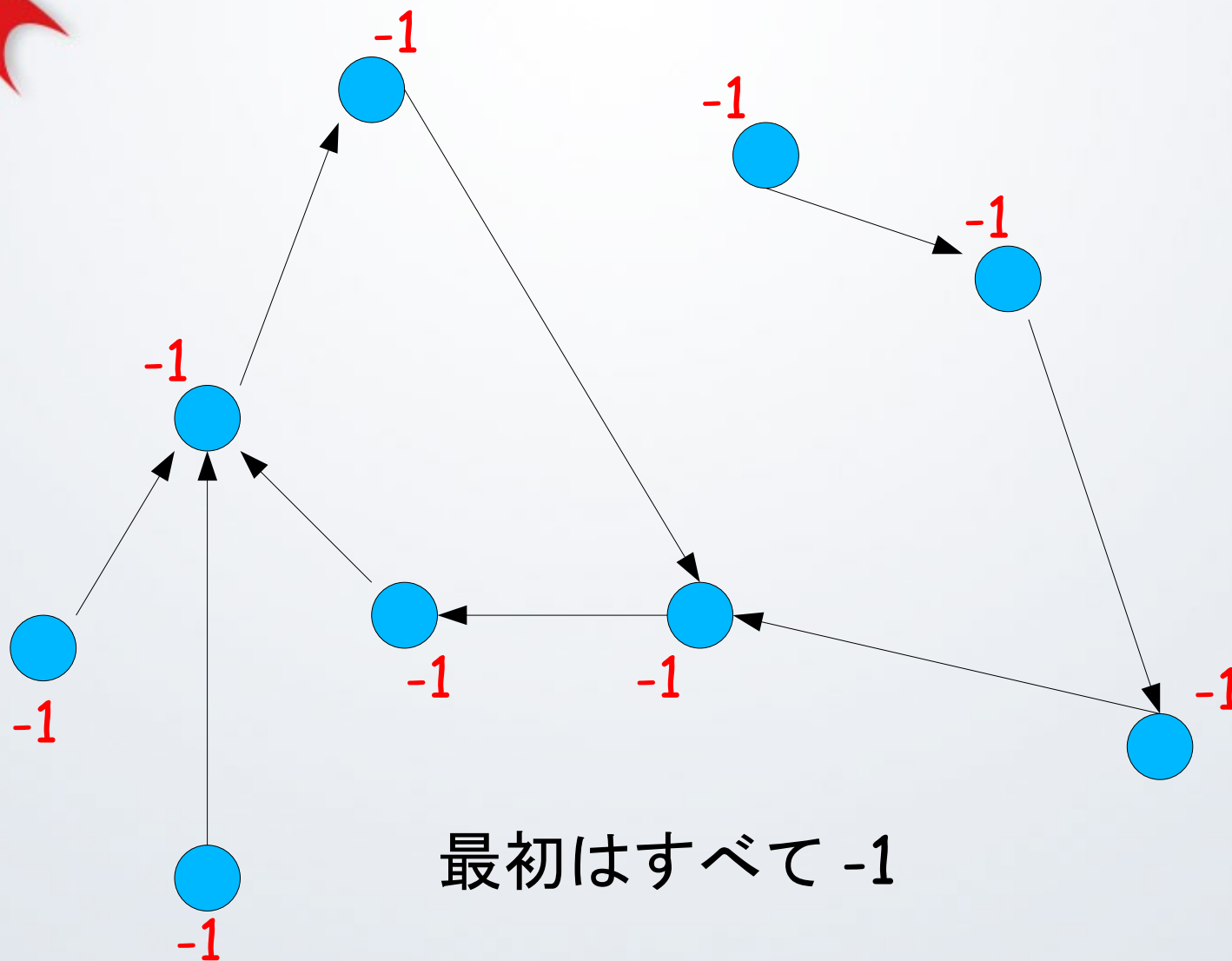
実装例  
4

## よさげな実装方法

- 実装例1(属性をつける)と同じような方法だが、属性ではなくたどり始めた頂点の番号をつける
- 最初この属性をあらわす数値はすべて  $-1$  にしておく

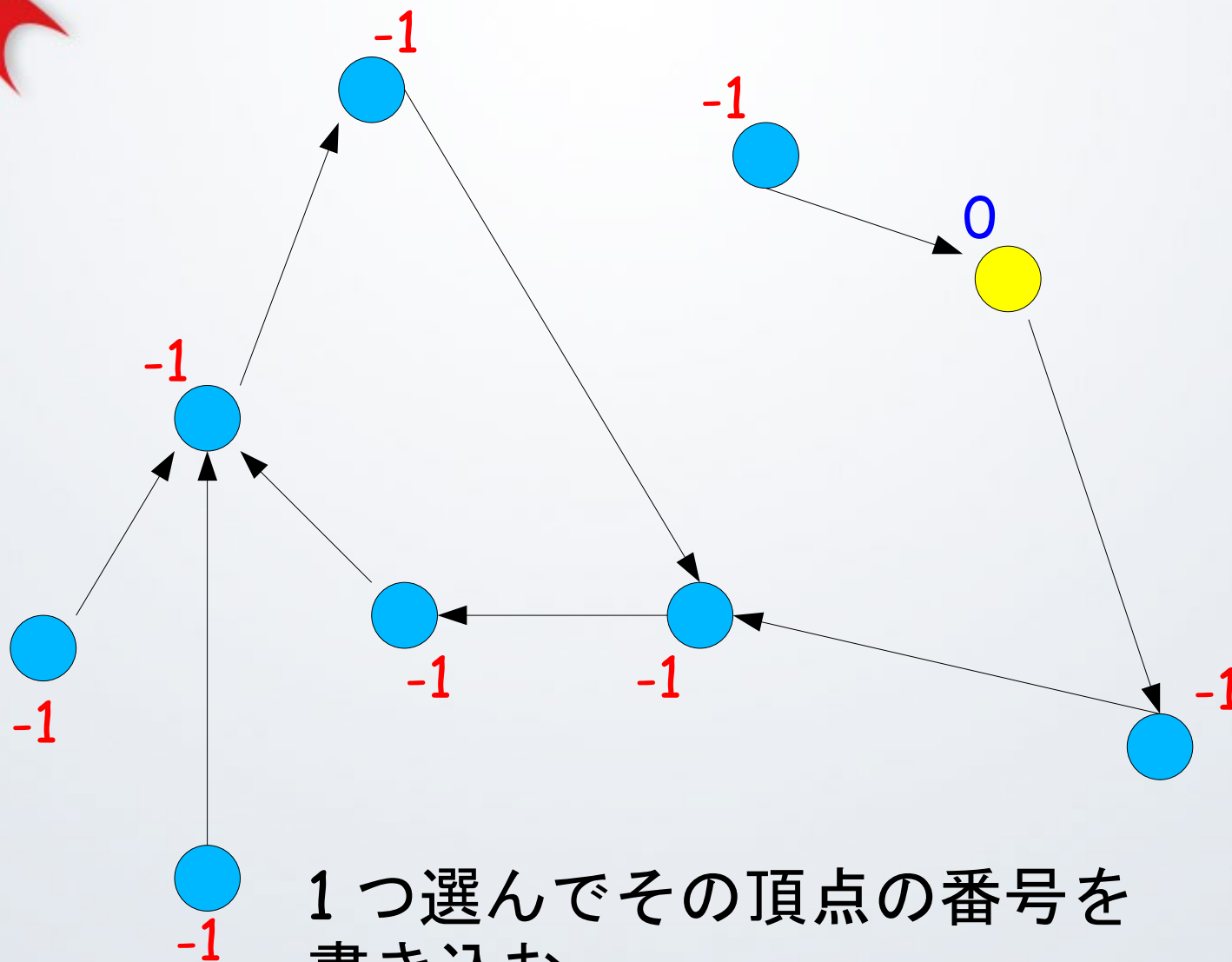
実装例  
4

# よさげな実装方法



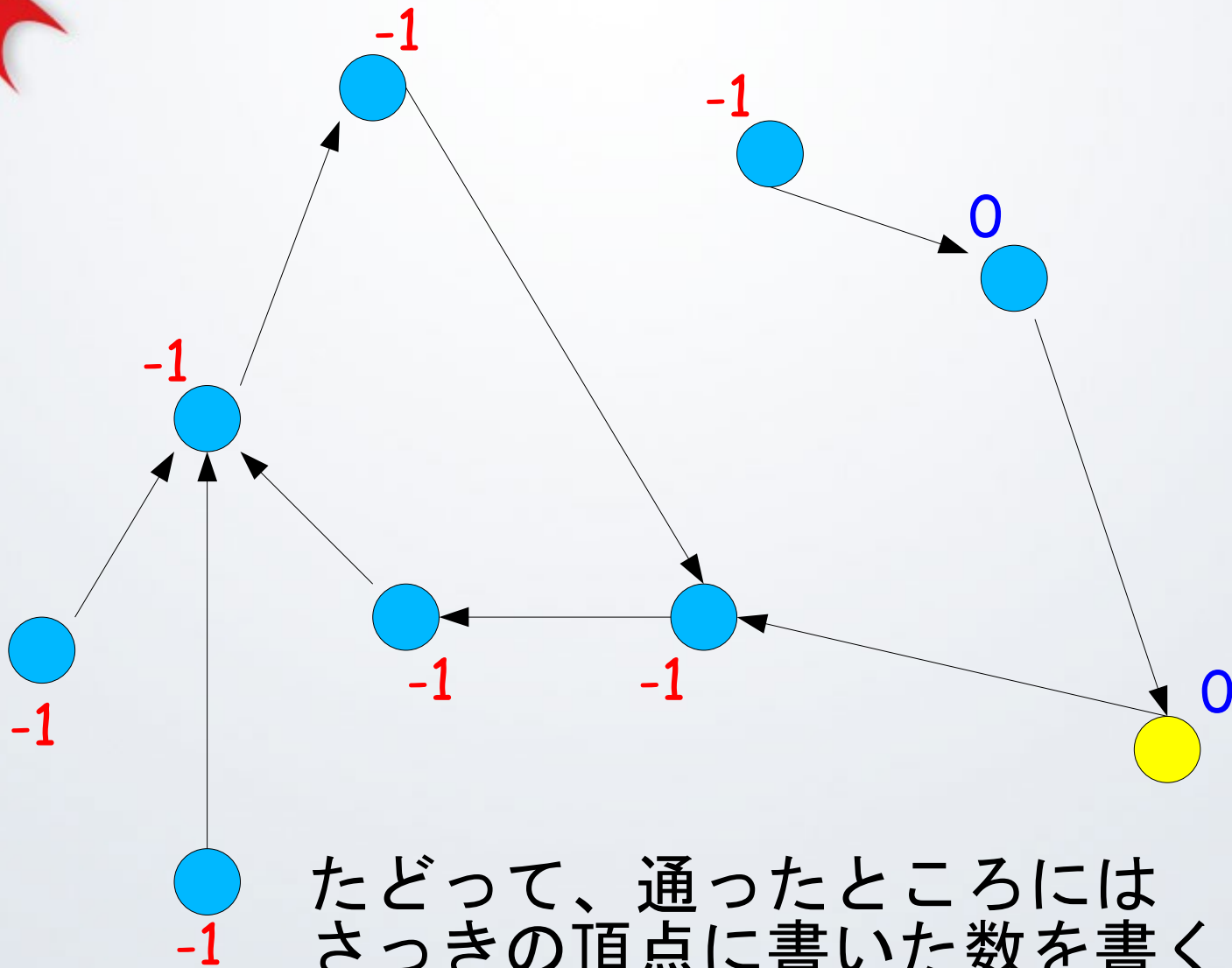
実装例  
4

# よさげな実装方法



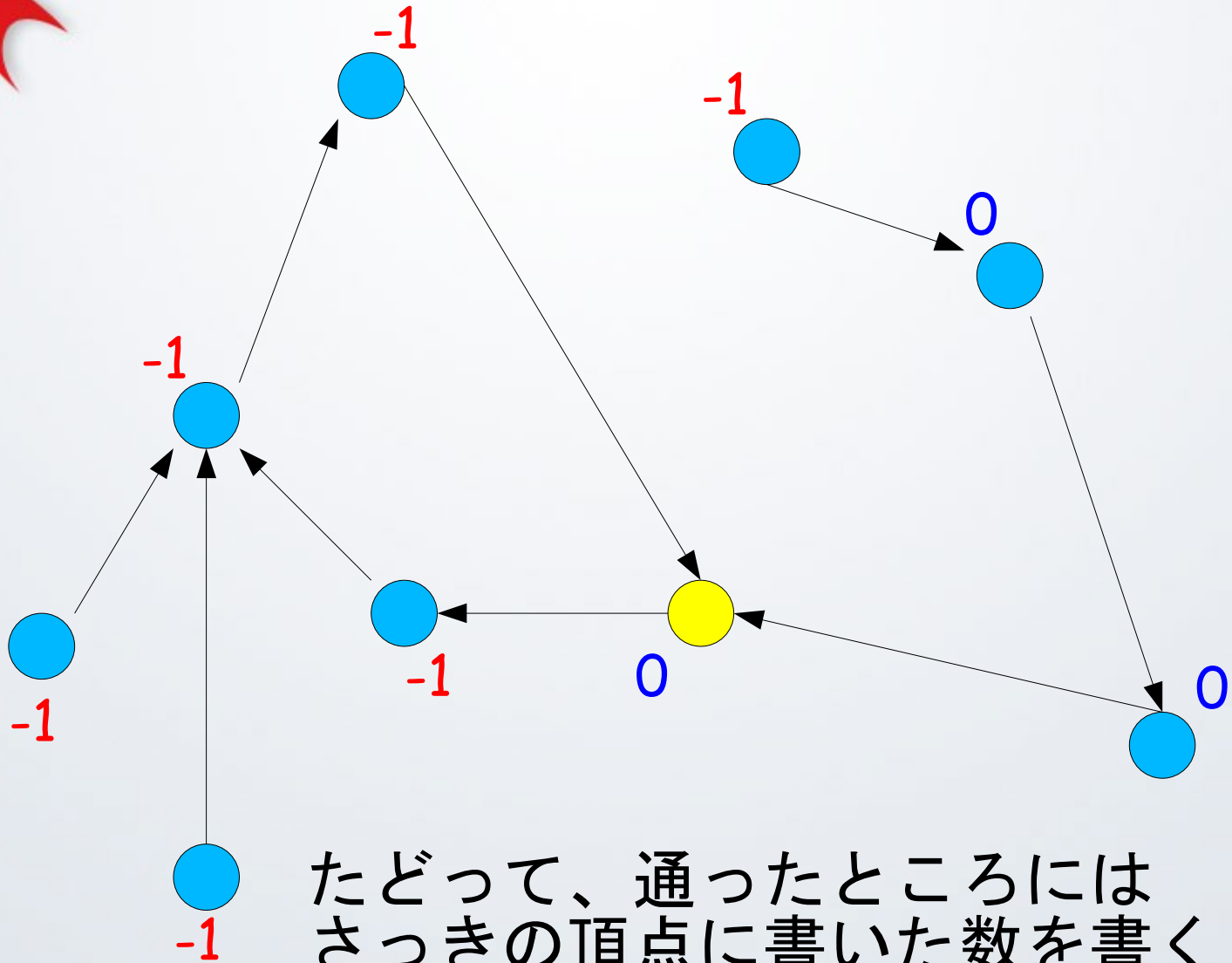
実装例  
4

# よさげな実装方法



実装例  
4

# よさげな実装方法

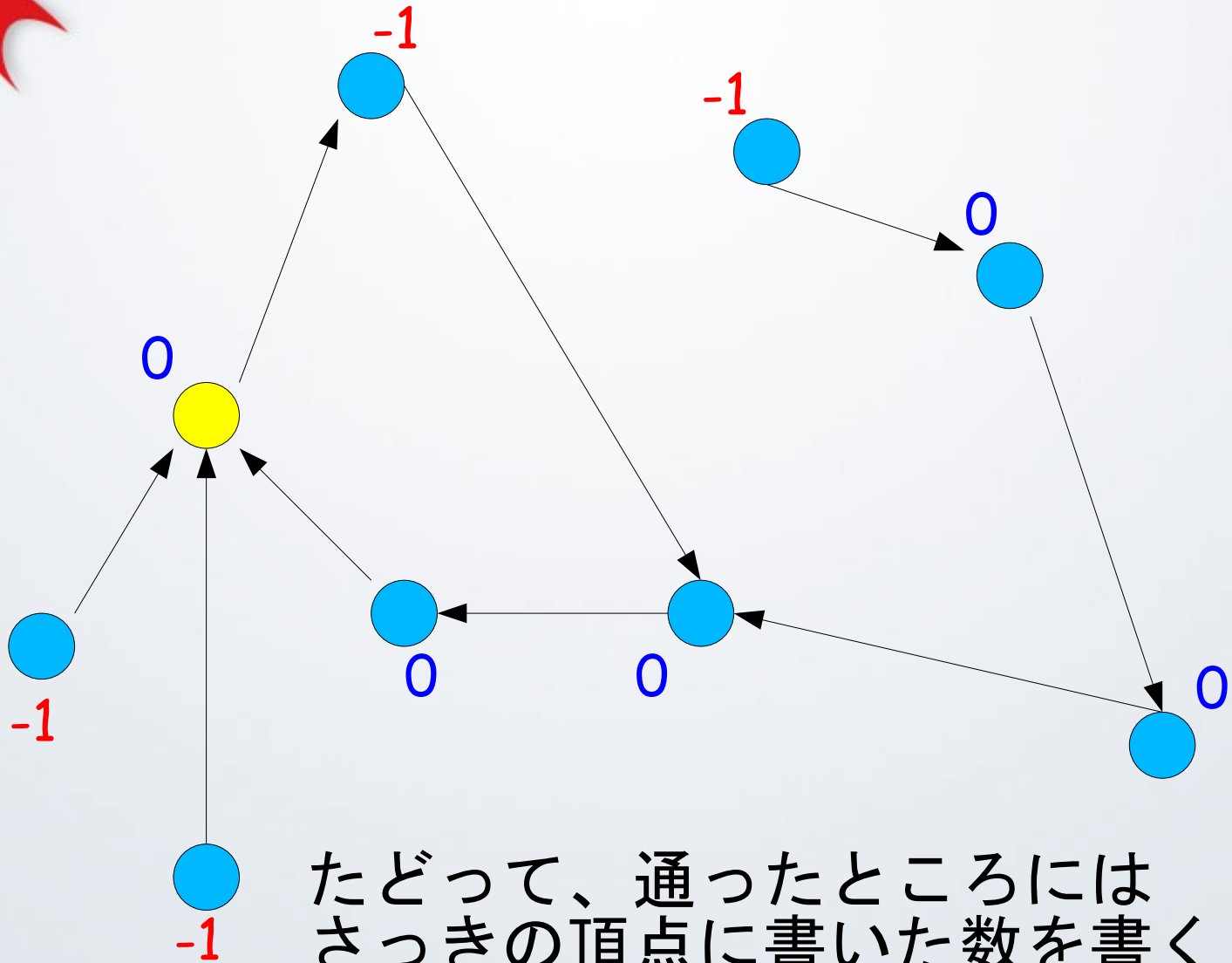






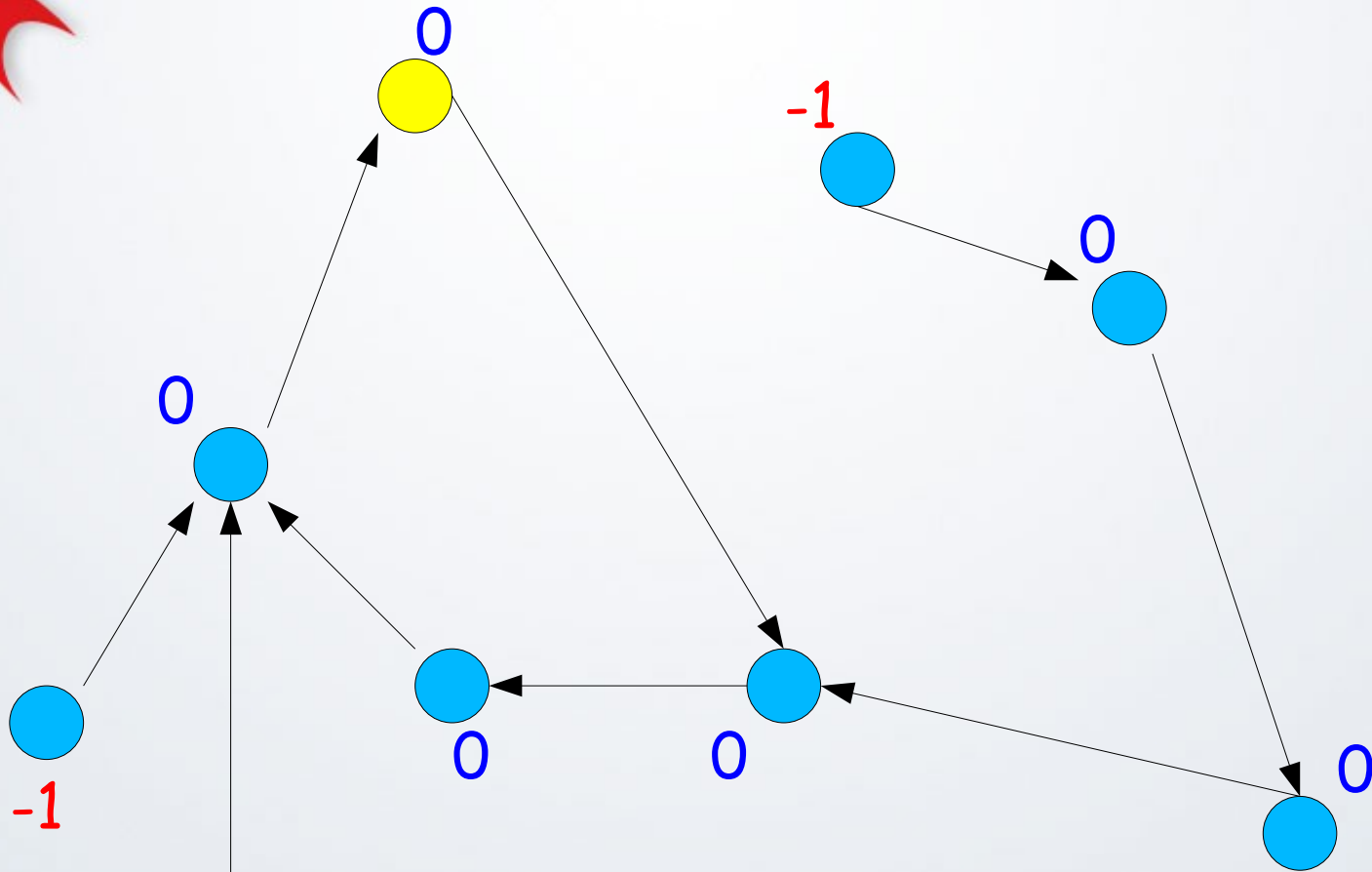
実装例  
4

# よさげな実装方法



実装例  
4

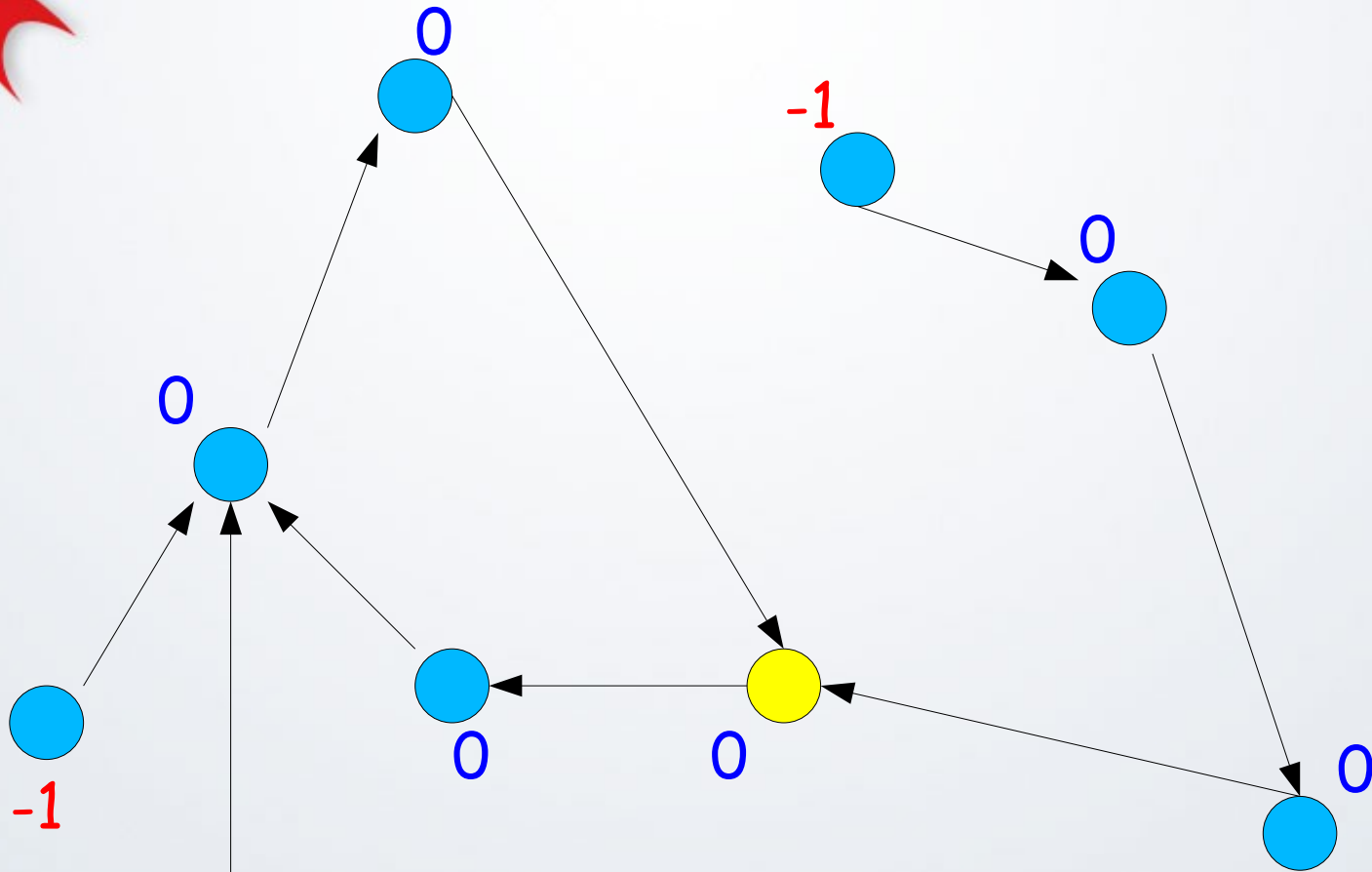
# よさげな実装方法



たどって、通ったところには  
さっきの頂点に書いた数を書く

実装例  
4

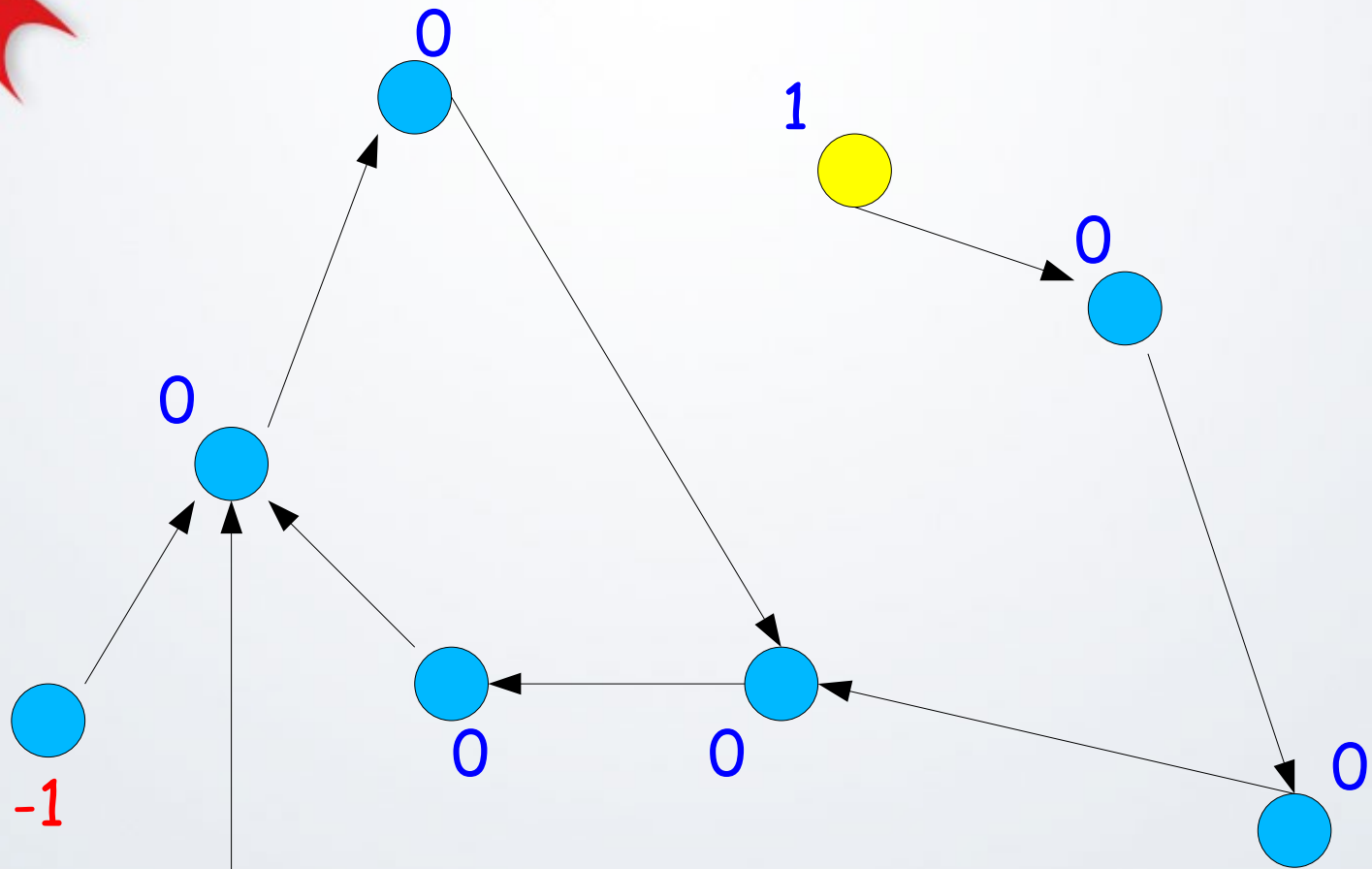
# よさげな実装方法



● -1 でない値を発見  
-1 最初に書いた数と同じだ→これはサイクル

実装例  
4

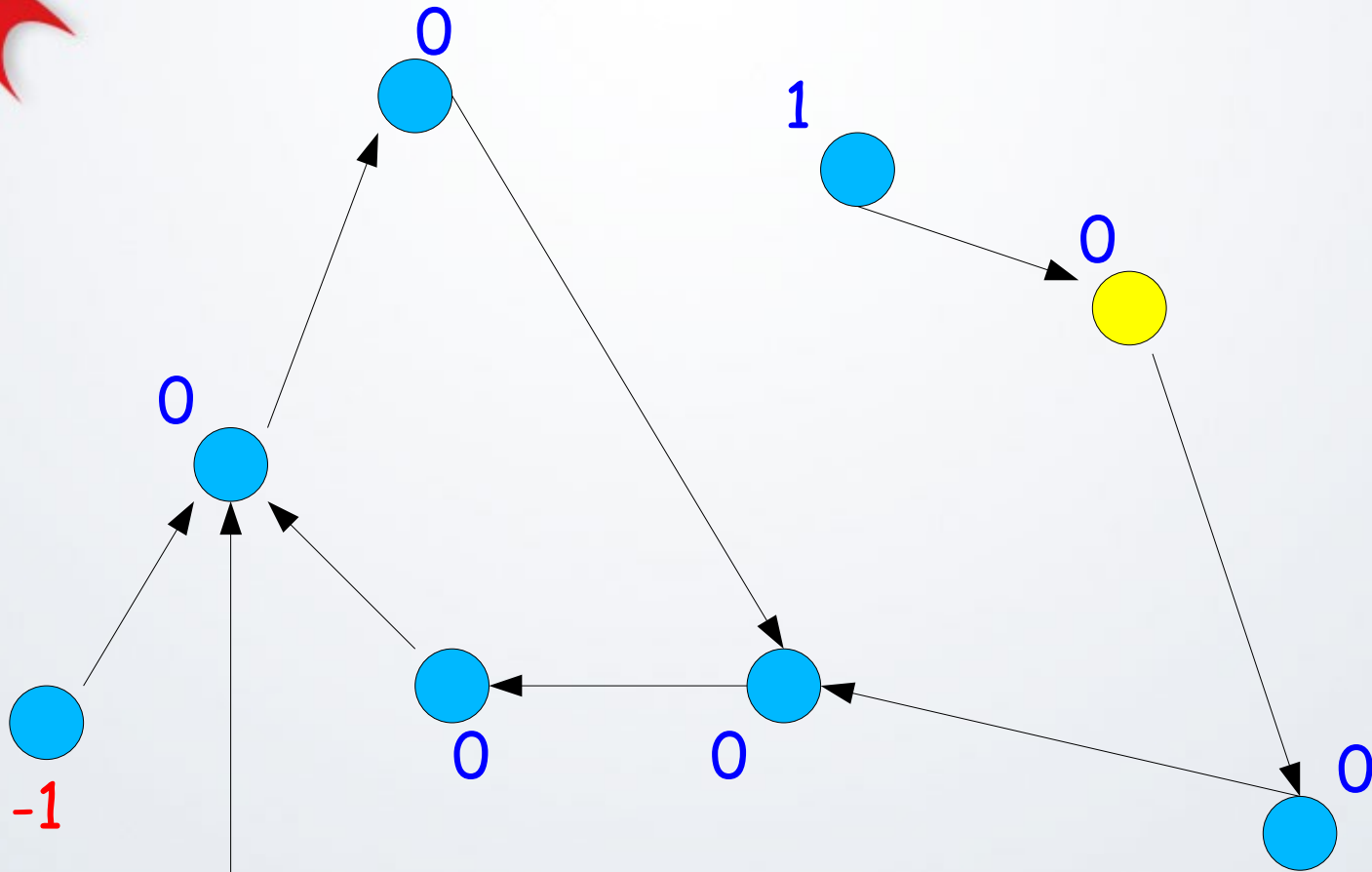
# よさげな実装方法



別の頂点でやってみる  
-1

実装例  
4

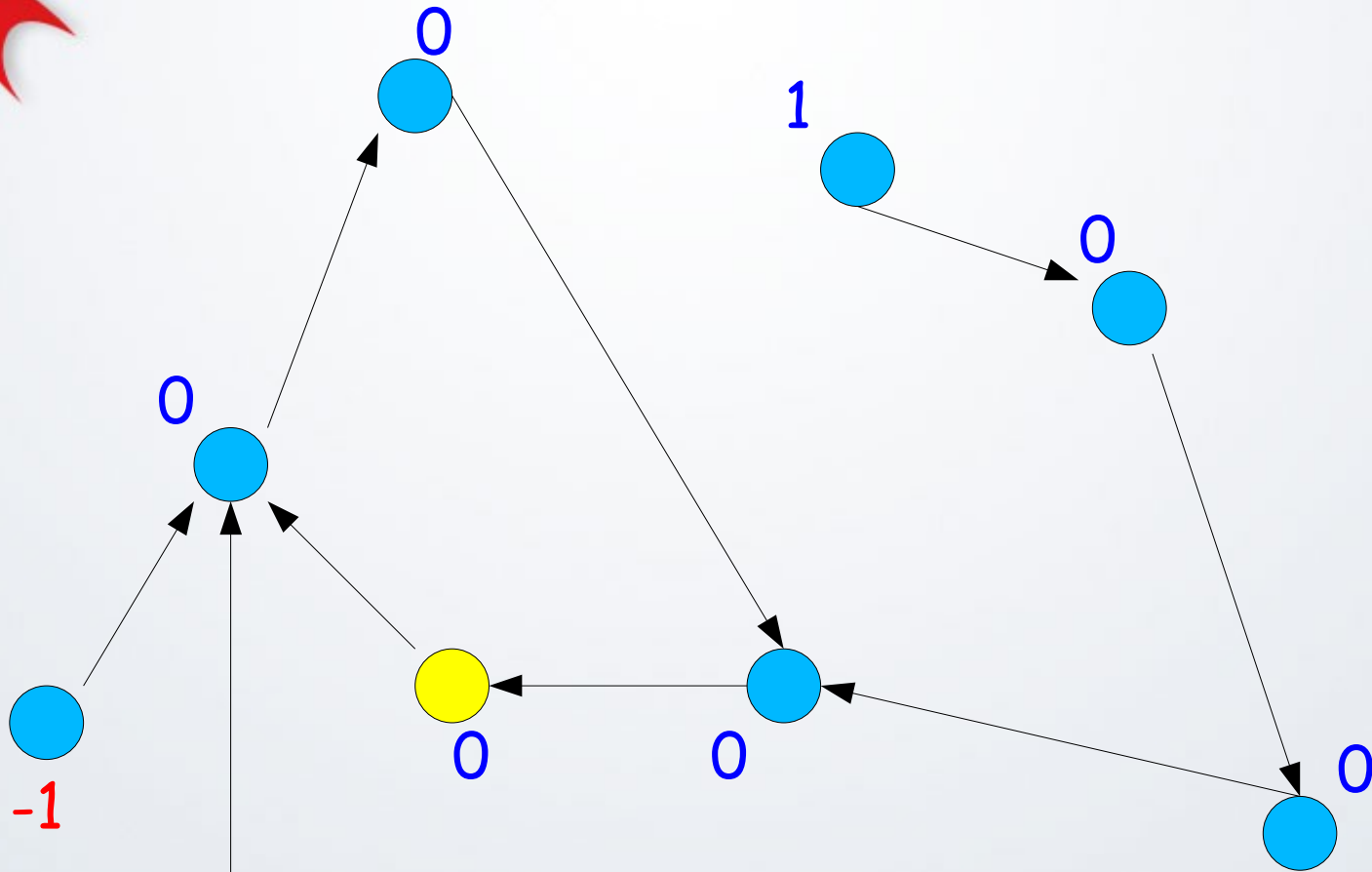
# よさげな実装方法




-1 ではないが、最初書いたのと違う数字に出会う → サイクルではなかった

実装例  
4

# よさげな実装方法



つぎはここ →いきなり書いてあるのです  
-1 でに訪れた (のこりは省略)



実装例  
4

## よさげな実装方法

- サイクルを検出する作業とそれぞれのサイクルに関する作業はまとめてしまったほうが楽（どのサイクルを処理したかわかりやすい）
- サイクル以外の辺に関する作業はどんな順番でも求められるのでその都度できる
  - サイクル検出と最大値を求めるのを同時に行うと実装が楽





## 練習問題

- 木 + 辺 1 本のグラフが出てくるもの
  - Report (JOI2011 春合宿)
  - ISLANDS (IOI2008)
  - Cthulhu (CF#80 Div2C & Div1B)
- 精神攻撃
  - パーティー (JOI2009 予選 3)

入力例 2 において, あなたには友達はいない. したがって, あ



## 余談

- この問題を一般のグラフについて考えてみよ  
うとしたけど、負辺のあるフローが効率的に  
求められないらしいので冷えてます
- 何か分かったら教えてください……

得点分布

0点いない! Great!!

