

# Problem I: Aaron と Bruce

問題作成者: 八森

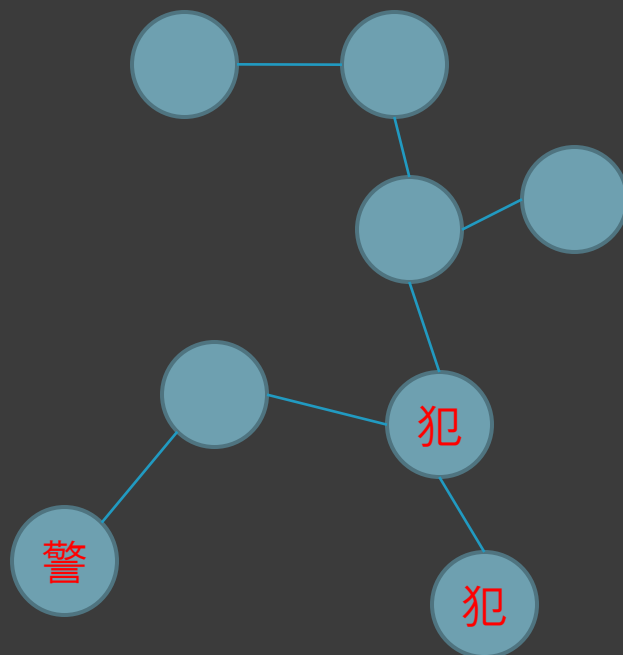
解法作成者: 八森、松本

発表: 八森

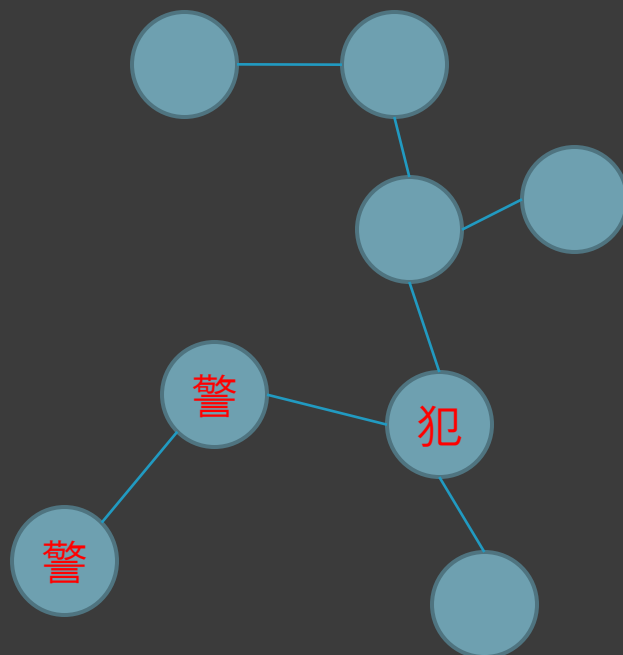
# 問題概要

- ◎ AaronとBruce はグラフ上を追いかけっこする
  - ◎ AaronはBruce からできるだけ長く逃げ回りたい
  - ◎ BruceはAaron をできるだけ早く捕まえたい
- 
- ◎ Aaronがつかまるのにかかる時間は？

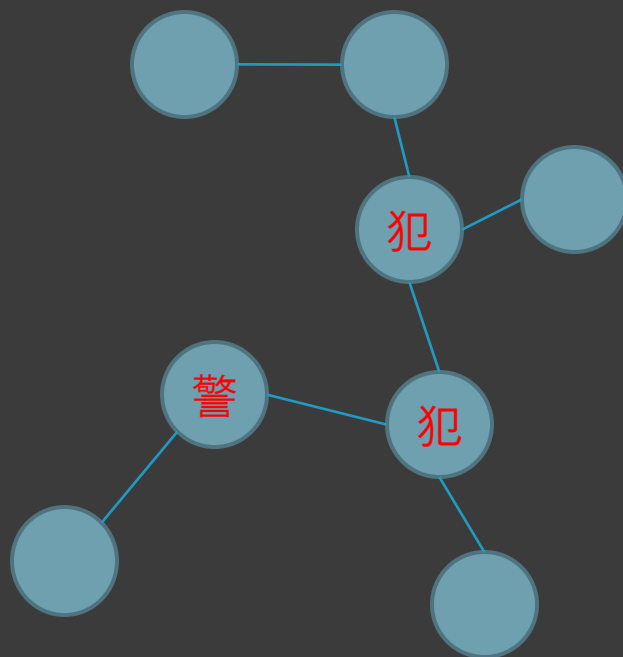
# サンプルの例



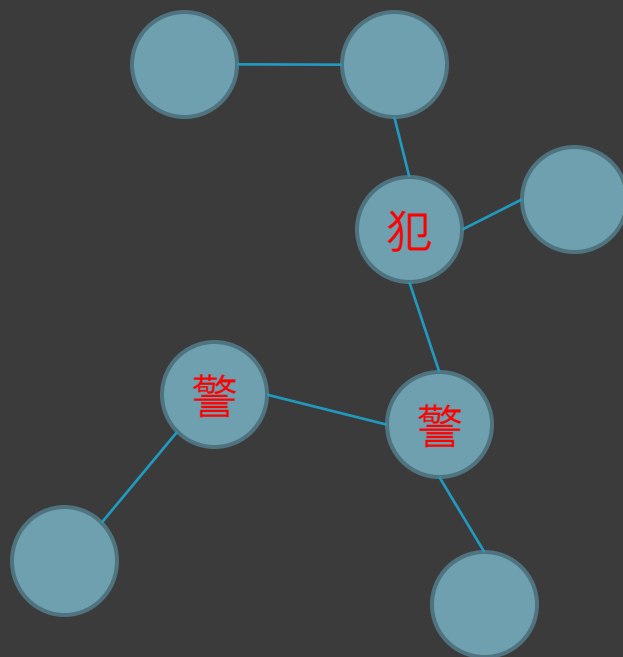
# サンプルの例



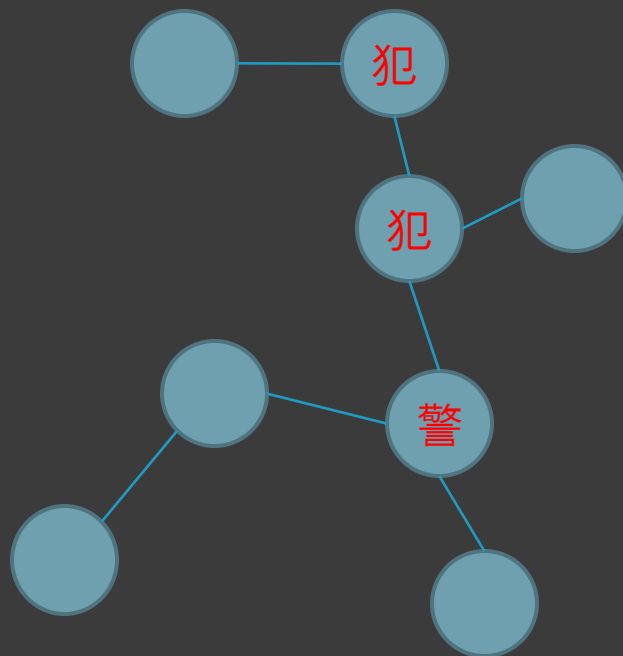
# サンプルの例



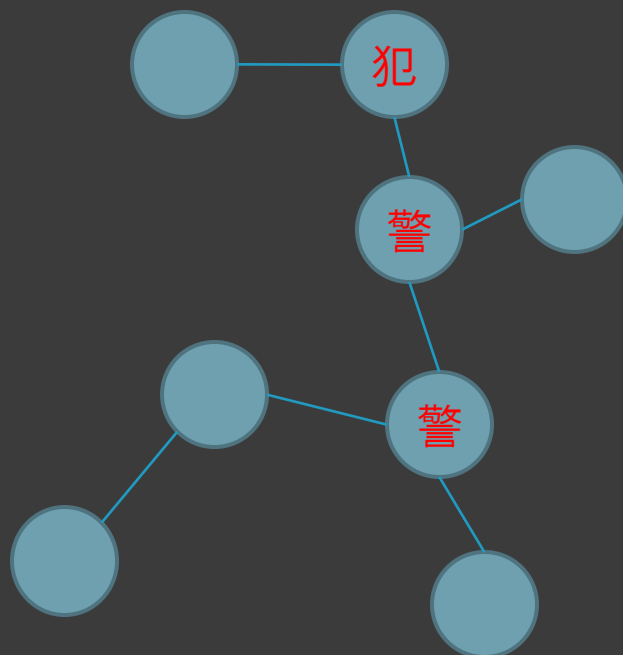
# サンプルの例



# サンプルの例

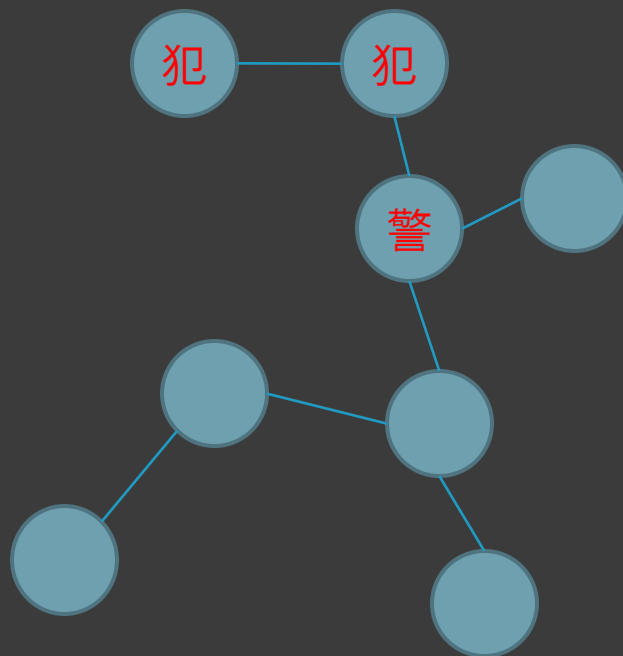


# サンプルの例

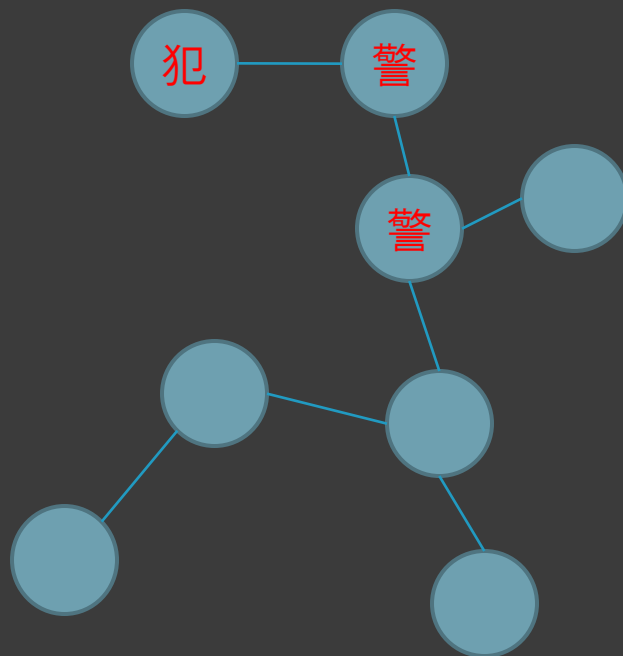




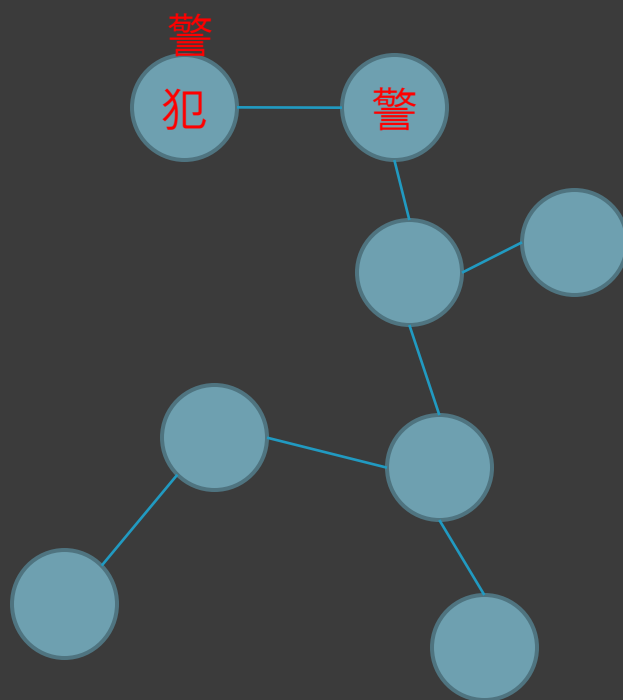
# サンプルの例



# サンプルの例



# サンプルの例



# 解法

- ◎ 方針:

AaronとBruceが同じノードにいる時を終了状態とし、その状態から直前に二人がいる位置を復元していく。

状態は(どっちのターンか、Aaronの位置、Bruceの位置)で表される。

# 解法

- ◎ 3次元配列 `table[2][50][50]` を用意。

`table[turn][a][b]:`

現在(Aaron|Bruce)の番。Aaronがaにいて、Bruceがbにいるとき、AaronがBruceにつかまるのにかかる時間。

AaronがBruceにつかまったときのtableの値は0で初期化。

(`table[Aaronのターン][n][n] = 0`)

# 解法

## ◎ Aaronの番のときの処理

Aaronがどのノードに行くのが最適か調べる。

$\text{table}[\text{Aaronのターン}][a][b] := \max(\text{table}[\text{Bruceのターン}][a'][b])$

$a'$  :  $a$  に隣接するノード

# 解法

## ◎ Bruceの番のときの処理

Bruceがどのノードに行くのが最適か調べる。

$\text{table}[\text{Bruceのターン}][a][b] := \min(\text{table}[\text{Aaronのターン}][a][b'] + 1)$

$b'$ :  $b$  に隣接するノード

# 注意

- ◎ tableの値がノード数を超えたら、tableの値をinfinityとする。
  - BruceがAaronを捕まえられるならば、捕まるのにかかるターン数は多くとも(ノード数-1)となるから。
- ◎ tableの値が収束するまで計算を繰り返す。
- ◎ tableに値が入っていないときはそこを参照しない。



# 計算時間

$O(\text{収束にかかるターン数} * \text{状態数} * \text{隣接するノード数})$

-> $O(50 * 2 * 50 * 50 * 50)$

-> $O(1250\text{万})$

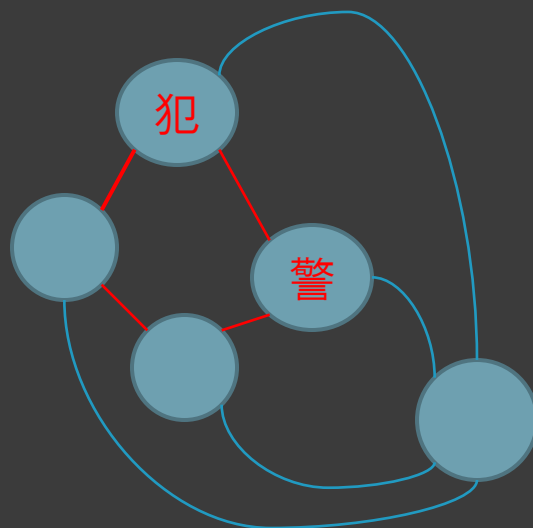
# だめな解法 その1

- ◎ AaronとBruceの初期状態からメモ付探索
  - 計算が終わってない部分を参照するので正しい答えを出さない。

# だめな解法 その2

- ◎ サイクルの長さが4以上の部分を探してinfinity検出
- AaronとBruceがサイクルの上においてもAaronが逃げ続けられるとは限らない。

◎ たとえば、



**警**が右下のノードに移動すれば、**犯**は  
どうあがいても次のターンで捕まる。

# Submission状況

- ◎ 送信者数: 6
- ◎ 総送信数: 11
- ◎ 正解数: 4
- ◎ 最速正解者: 198分 片岡さん