

# JAG ICPC模擬地区予選2021

## J: Isomorphic?

---

原案: not

問題文: climpet

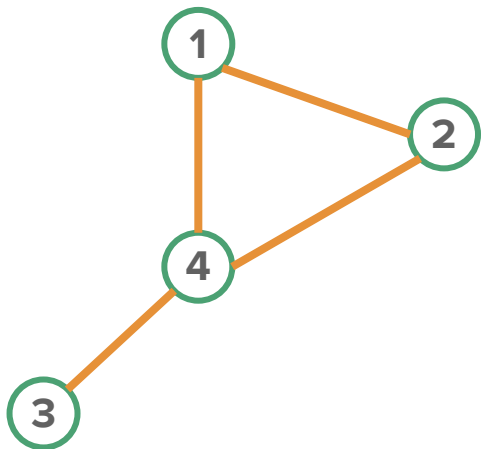
データセット: riantkb

解答: beet, climpet, hos, riantkb

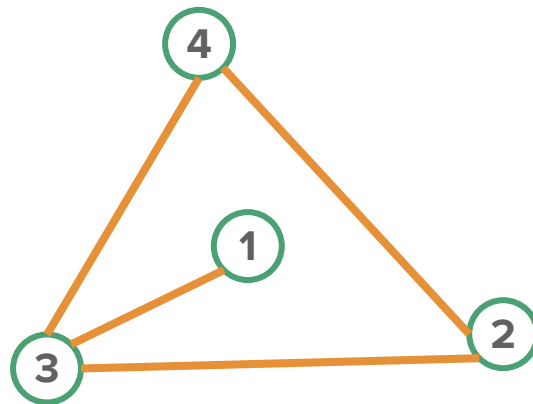
解説: riantkb

# 問題概要

- N 頂点 N 辺の単純連結無向グラフが 2 つある
- この 2 つのグラフが同型かどうか判定せよ

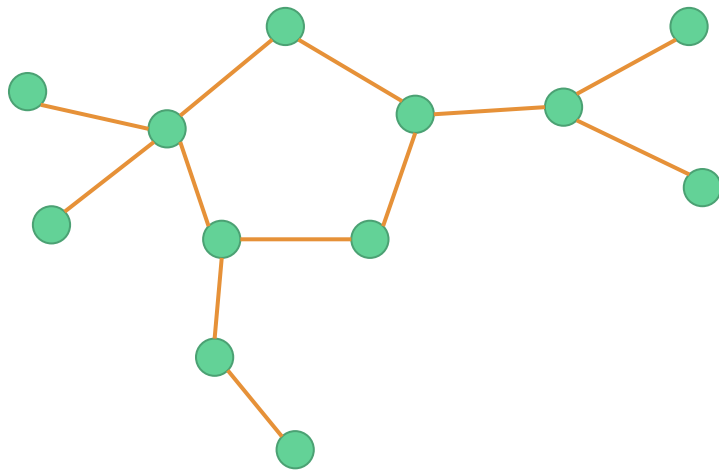


この 2 つのグラフは同型



# はじめに

- $N$  頂点  $N$  辺の連結無向グラフは、ちょうど一つサイクルがありそこから木が複数生えているような形をしている
  - このようなグラフを **Unicyclic graph** と呼んだり、日本語で**なもりグラフ**と呼ぶことがある

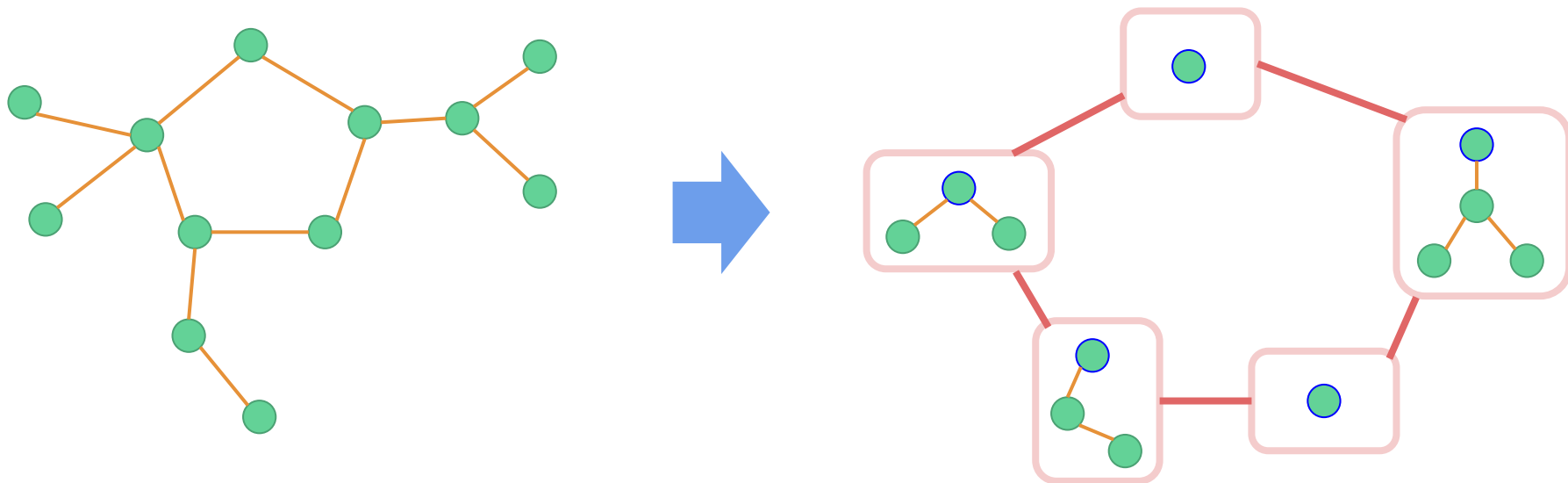


# はじめに

- 2つの根付き木が同型かどうか、という問題にはいくつか解決方法があり、例えば AHU アルゴリズムや根付き木に対するハッシュを設計する方法がある
  - ここでは詳しく解説しないので、詳しく知りたい方は例えば以下の記事をご覧ください
- AHU アルゴリズムについて
  - <https://chocobaby-aporo.hatenablog.com/entry/2017/12/05/233027>
- 根付き木に対するハッシュ設計について
  - <https://rng-58.blogspot.com/2017/02/hashing-and-probability-of-collision.html>

# 考察

- 与えられるグラフはサイクルに木が生えた形をしているため、それぞれの木をハッシュ等に変換しておくことで、それらの列が同じかどうかで元のグラフが同型であるかどうかを判定できると考えられる



## 考察

- サイクルを切り開いて列として考えると、それぞれのグラフから得られたハッシュ列 A, B に対する以下の問題の答えが本問題の答えと同じであると考えられる

- 要素列 A, B が与えられる。A に以下の操作を好きな順序で好きなだけ行い、A を B に一致させられるか？
  - A 全体を rotate する (  $a_1, a_2, \dots, a_N \rightarrow a_2, a_3, \dots, a_N, a_1$  )
  - A 全体を reverse する (  $a_1, a_2, \dots, a_N \rightarrow a_N, a_{N-1}, \dots, a_1$  )

# 考察

- 前述した問題は例えば A, B, B という順で連結した列に対し z-algorithm を適用したり、Rolling Hash などを用いることで解くことができる
  - 反転操作を考慮するため、A または B を反転したものについて再度判定する必要があることに注意

# 解法

- 与えられたグラフをサイクルとそれに繋がっている木、という形に整理し、それぞれの木についてハッシュ等を求める
- それらをサイクルの順に見ることでハッシュ列を得て、2つのハッシュ列がサイクルで見ても一致しているかを z-algorithm や Rolling Hash などで判定する
- 計算量は  $O(N)$  または  $O(N \log N)$  などとなる



## ジャッジ解

- beet (C++): 162 lines, 3.2 kB
- climpet (C++): 221 lines, 4.2 kB
- hos (C++): 170 lines, 3.9 kB
- riantkb (C++): 174 lines, 4.7 kB

# 統計情報

- Acceptances / Submissions
  - 25 / 113 (22.12 %)
- AC teams / Trying teams
  - 25 / 29 (86.21 %)
- First Acceptance
  - SPJ (30 min)