

ICPC Japan Alumni Group

夏合宿 2015 Day2

G: Escape

原案・問題文: zerokugi

解答: zerokugi, rng_58,
sugim48

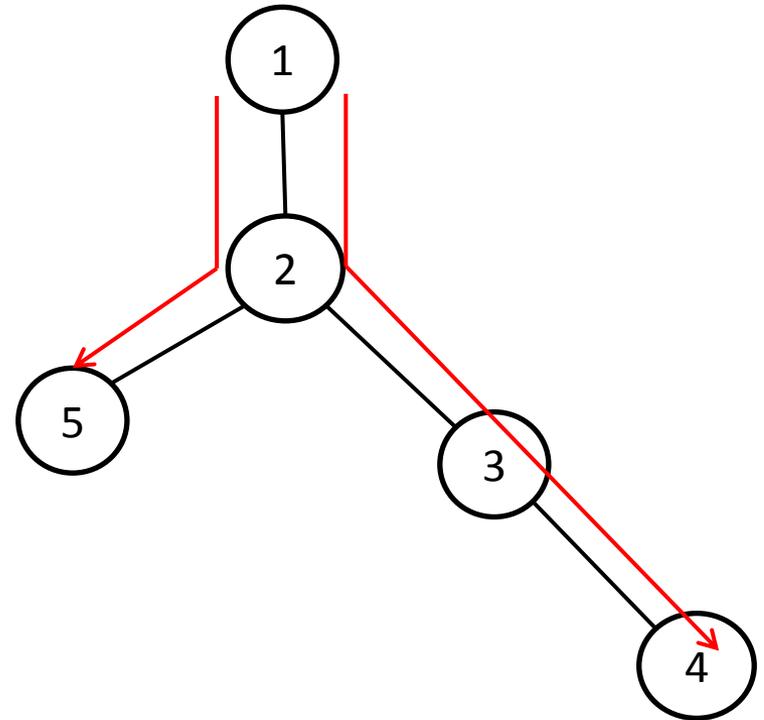
解説: zerokugi

問題概要

- 各頂点が正の値を持つ無向グラフが与えられる
- 頂点1を始点として「直前に通った辺を引き返すことができない」という制約のもとでグラフ上を移動することができる
- 通った頂点の値の総和を最大化せよ

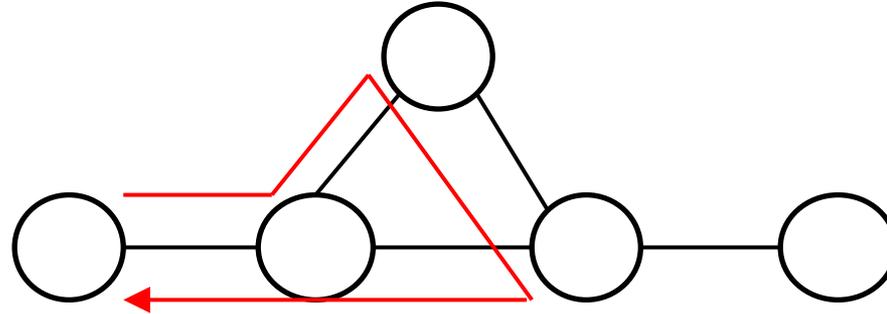
考察 (1)

- 木の場合
 - 根(頂点1)から葉へのパスの総和の最大値



考察 (2)

- サイクルがある場合
 - 引き返すことができる



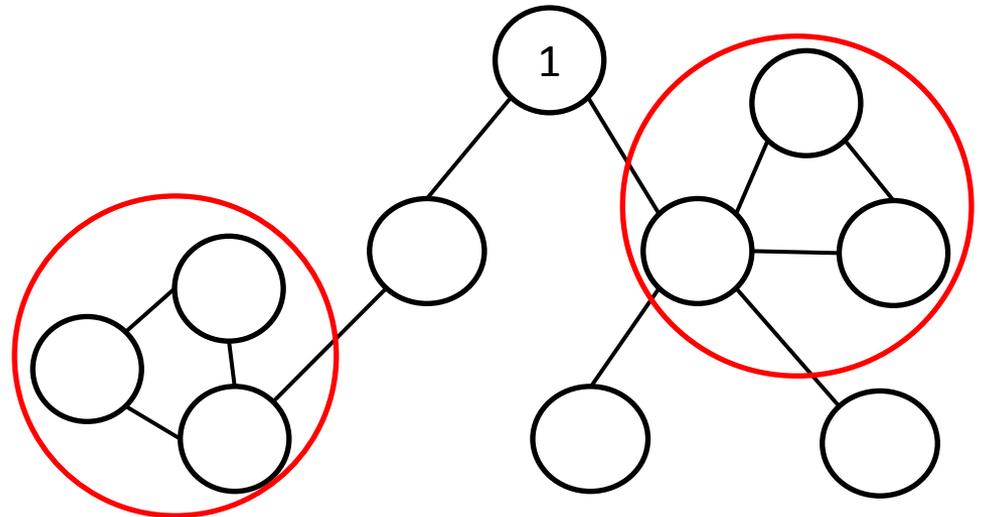
解法

- 解法1 二重辺連結成分分解 + 木DP
- 解法2 もうちょっと楽な方法

どちらも $O(N+M)$

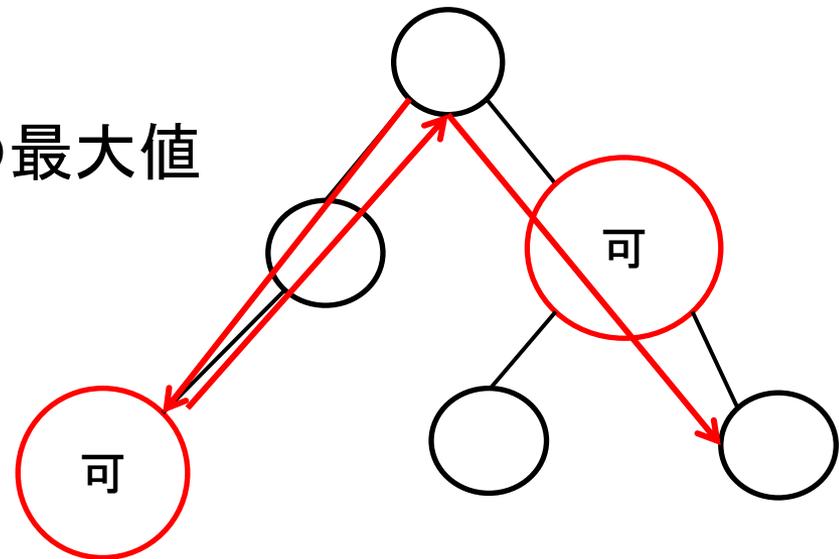
解法1

- 二重辺連結成分分解とは？
 - 任意の1辺を取り除いても連結が保たれるような頂点集合を一つの頂点としてまとめる
 - 要するにサイクル上の頂点がまとまってくれる
 - 分解後のグラフは木(非連結ならば森)になる



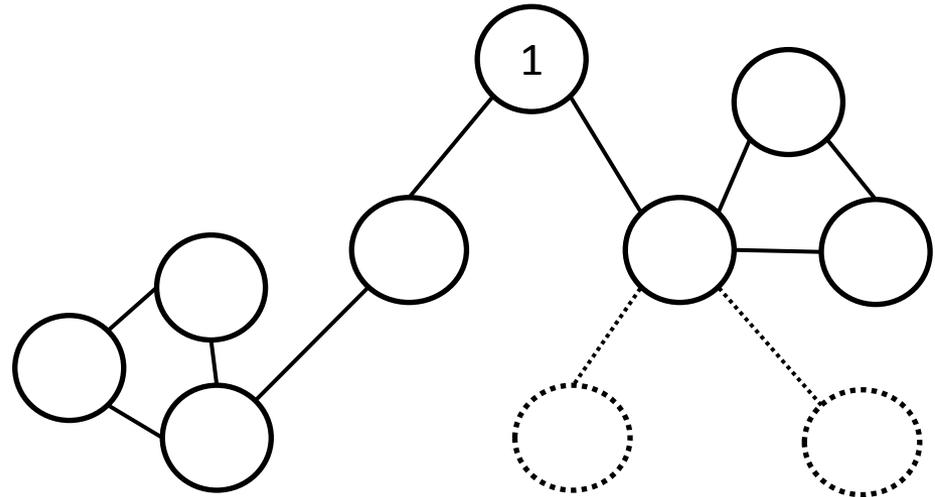
解法1

- 二重辺連結成分分解 + 木DP
 - 二重辺連結成分分解した結果、3頂点以上からなる成分は折り返し可能
 - 最後だけ折り返さずに葉まで進める
 - 各頂点について
 - 葉まで進む時の最大値
 - その頂点に戻ってくる時の最大値
- を木DPで計算



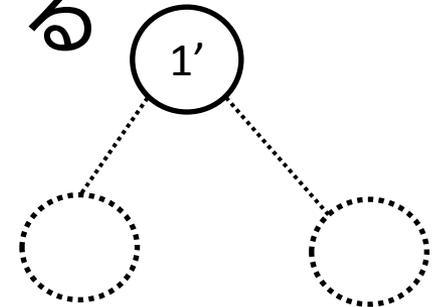
解法2

- 頂点1を除く次数1のノードを削ってみる
- 残ったノードは全て通ることができる
 - 2頂点以上ある場合、末端は必ずサイクルになるため引き返すことができる



解法2

- 残ったノードを頂点1にまとめて、消したノードを戻すと木になる
- 木の場合の解法 (考察 1) を適用する
- 多始点BFS 1回で可能
 - 実装がとても軽い
 - ICPC では待ち時間中に考察を進めておき、軽い実装方針を選択できるといいです



Writer 解 / 統計情報

- 解法 1
 - zerokugi: C++ 124行
- 解法 2
 - rng_58: C++ 98行
 - sugim48: C++ 49行
- FA: Cxiv-Dxiv (63:08)
- AC/Try : 14 / 23
- AC/Submission : 14/ 71