Two Currencies 解説

ynymxiaolongbao

クエリの言い換え

- X_j枚の金貨とY_j枚の銀貨がある
- 「金貨1払う or 銀貨C_i払う」という要求がいくつかある
- 金貨、銀貨の枚数を0未満にせずにすべての要求を満たせるか？満たせる場合、残せる金貨の枚数の最大値はいくつか？

- C_iが小さいものから順に、和がY_jを超えない限り銀貨を払う
- 残りを金貨で払う（足りなかったら-1）

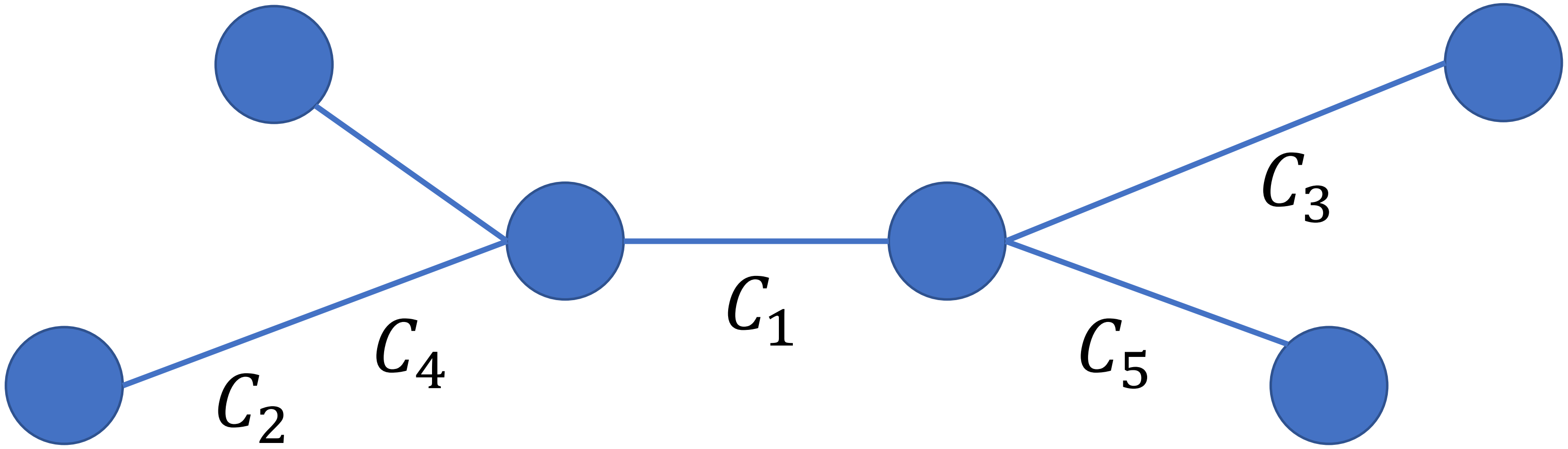
クエリの言い換え

- (答え)= $\max(X_j$ - (払いたい金貨の枚数) , $-1)$
- (払いたい金貨の枚数)
  =(要求の個数)- (C_iが小さいものから最大何個の和がY_j以下か)

以下の二つの値が分かれば良い

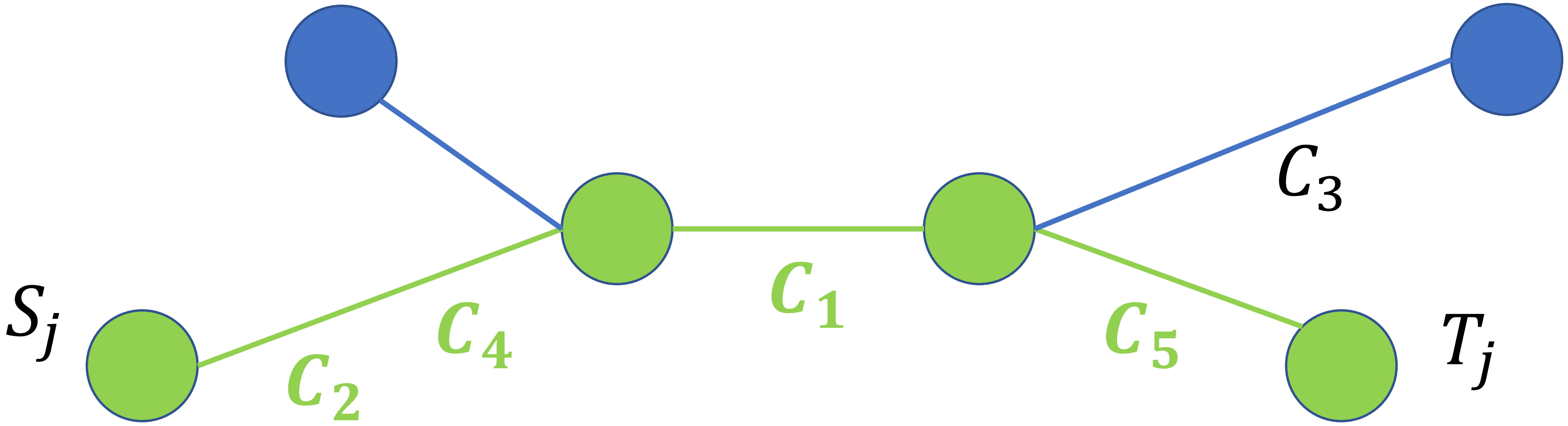
- (要求の個数)
- (C_iが小さいものから最大何個の和がY_j以下か)

問題概要



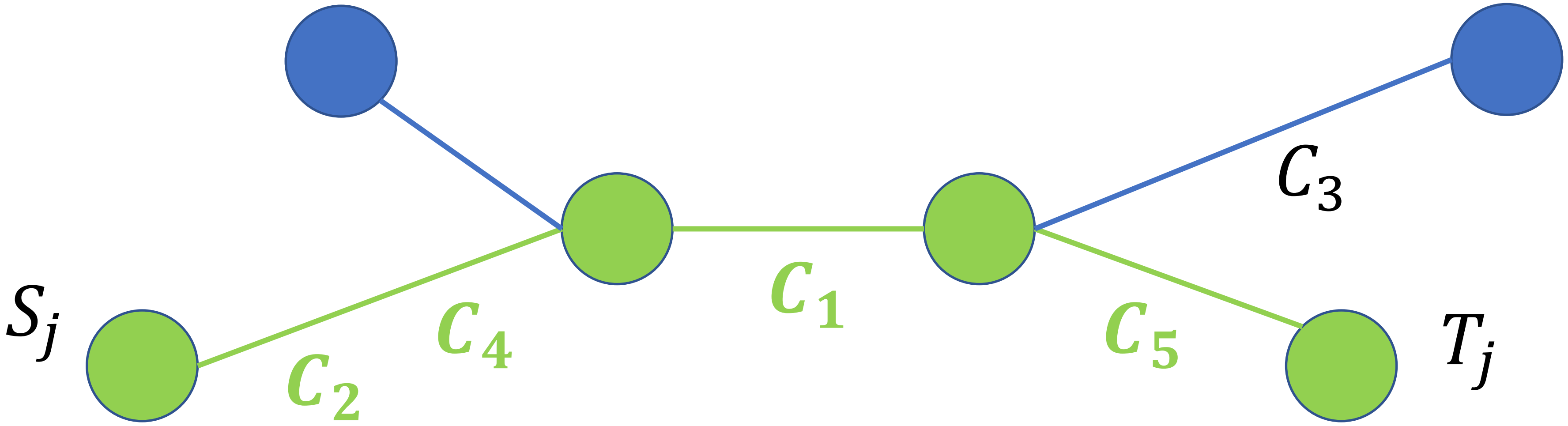
- 木がある
- 辺上に、値C_iがいくつかある
- 頂点S_j, T_jを結ぶ最短パス上の値C_iの集合について、（C_iの個数)、（C_iが小さいものから最大何個の和がY_j以下か）を求める

問題概要



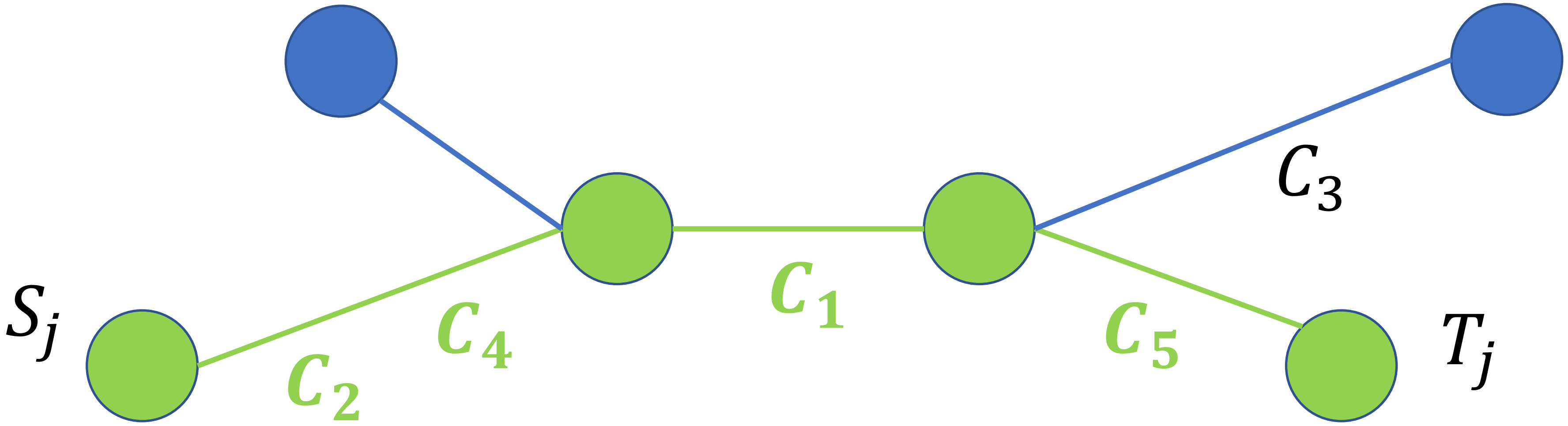
- 木がある
- 辺上に、値C_iがいくつかある
- **頂点S_j, T_jを結ぶ最短パス上の値C_iの集合**について、（C_iの個数）、（C_iが小さいものから最大何個の和がY_j以下か）を求める

小課題１（$N, M, Q <= 2000$）



- （C_iの個数）、（C_iが小さいものから最大何個の和がY_j以下か）求める
- 二つの値を愚直に求める
- $O((N + M \log M)Q)$

小課題 2　$(C_1 = C_2 = \cdots = C_M)$

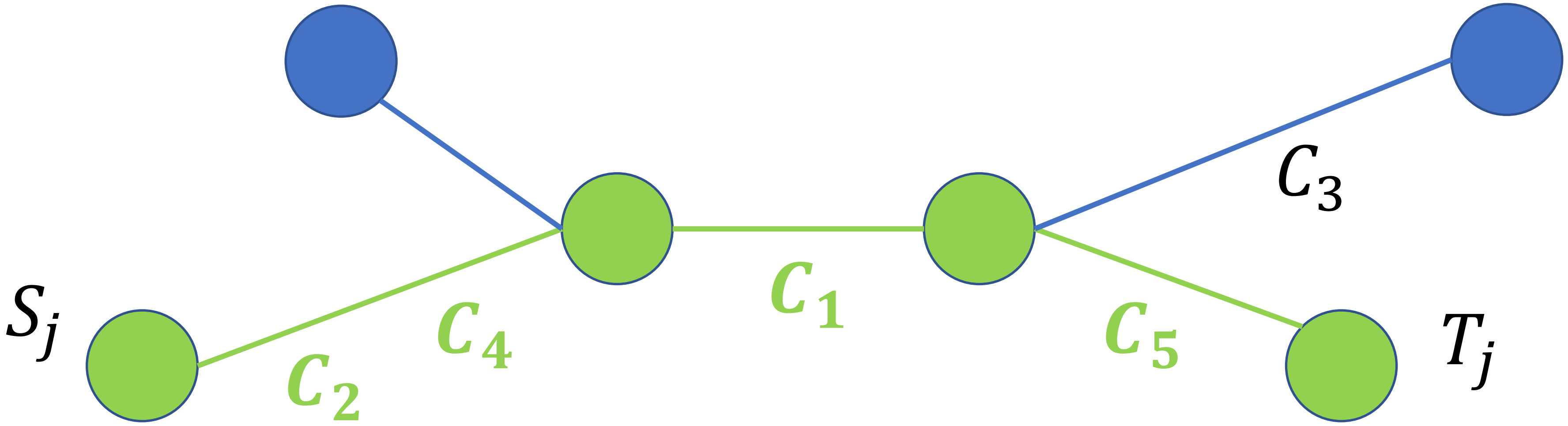


- (C_iの個数)、(C_iが小さいものから最大何個の和がY_j以下か) 求める
- (C_iが小さいものから最大何個の和がY_i以下か)

  $= \min\left((C_i\text{の個数}), \left\lfloor \frac{Y_j}{C_1} \right\rfloor \right)$

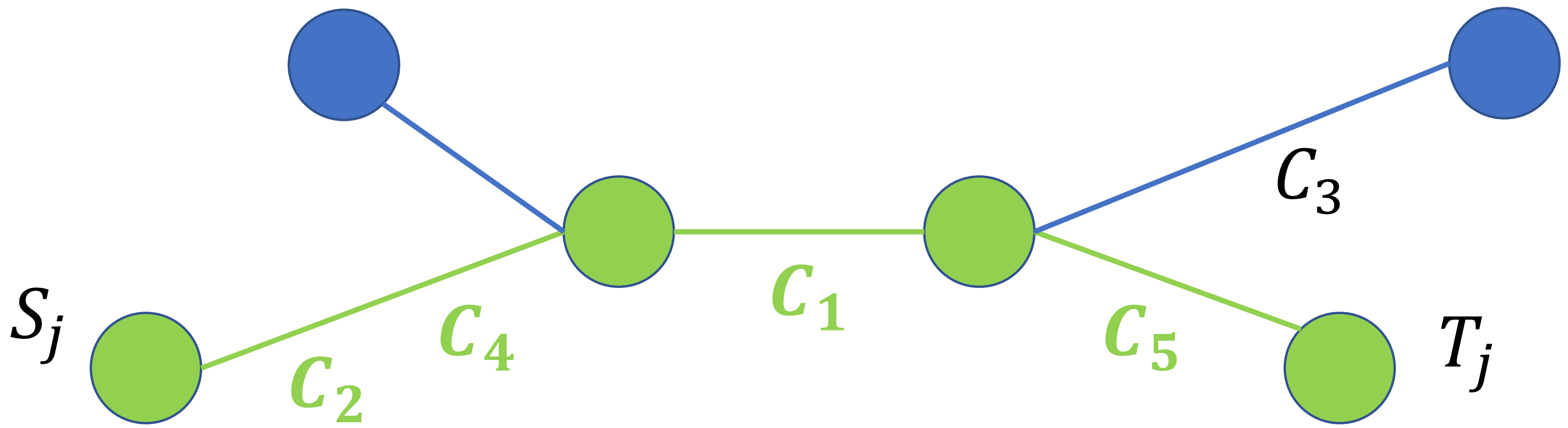
→ (C_iの個数) だけ求めれば良い

小課題２　$(C_1 = C_2 = \cdots = C_M)$



- $(C_i$の個数$)$ を求める
- 木のパス上の辺の重みの和（更新なし）
  →LCA＋累積和
- $O(Q \log N + N \log N + M)$

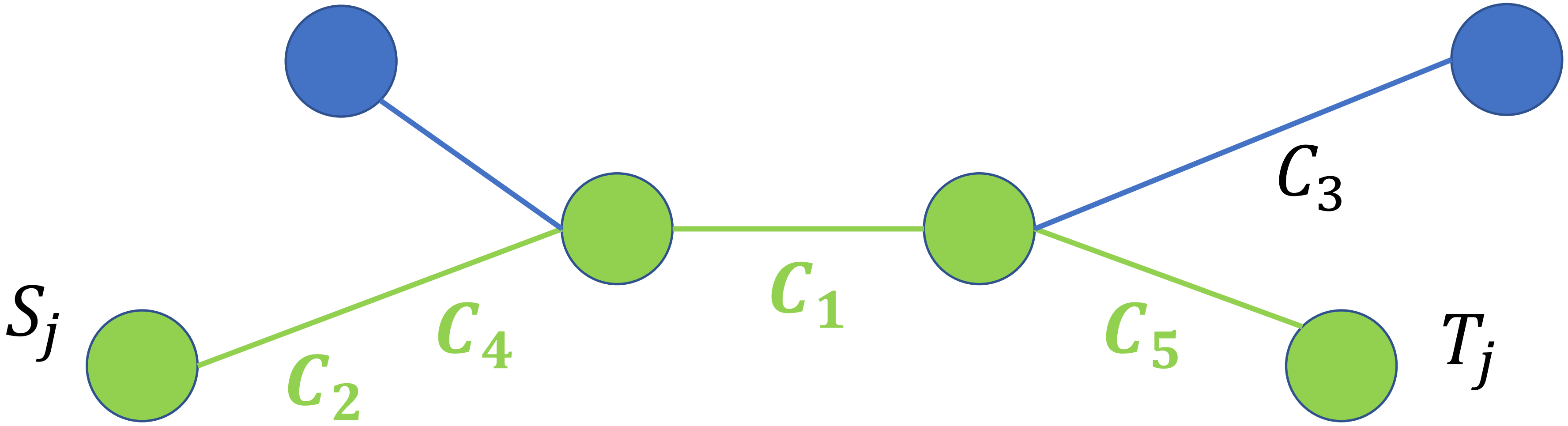
小課題３（パス）,満点解法



- (C_iの個数)、(C_iが小さいものから最大何個の和がY_j以下か) 求める
- (C_iの個数)は小課題２で求まった

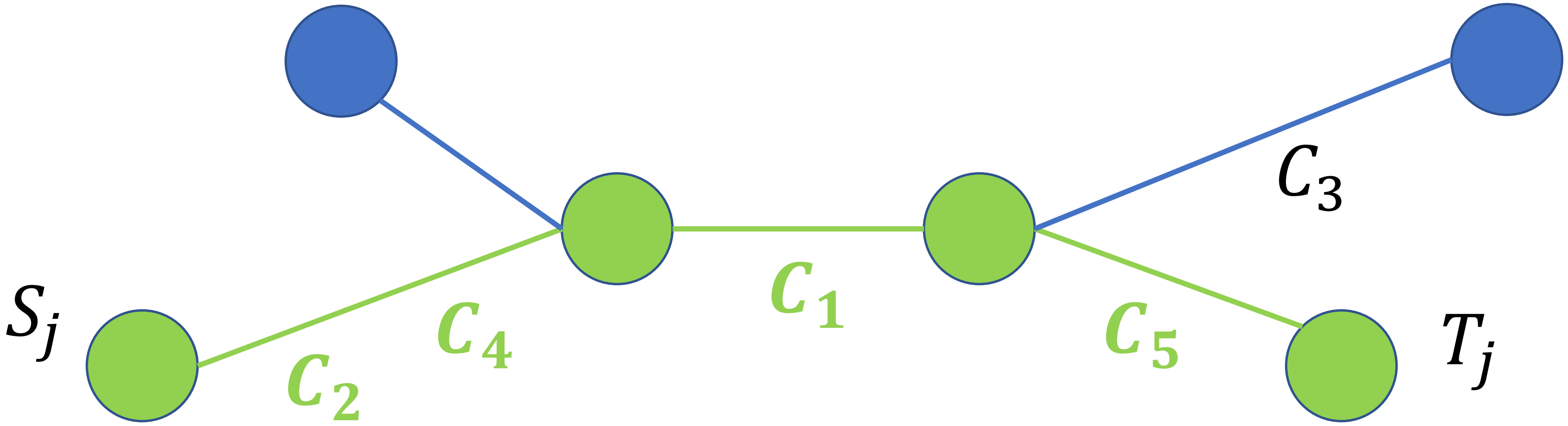
※もし木のケースが分からなくても、パスの場合は累積和で求まる
→ (C_iが小さいものから最大何個の和がY_j以下か) を求めればよい

小課題３（パス）,満点解法



- （C_iが小さいものから最大何個の和がY_j以下か）を求める
- パス$S_j - T_j$上にあるC_iであって、全てのC_iを昇順にソートしたときk番目までに入るものの和がY_j以下となる最大のkは何か？
- このままでは難しい

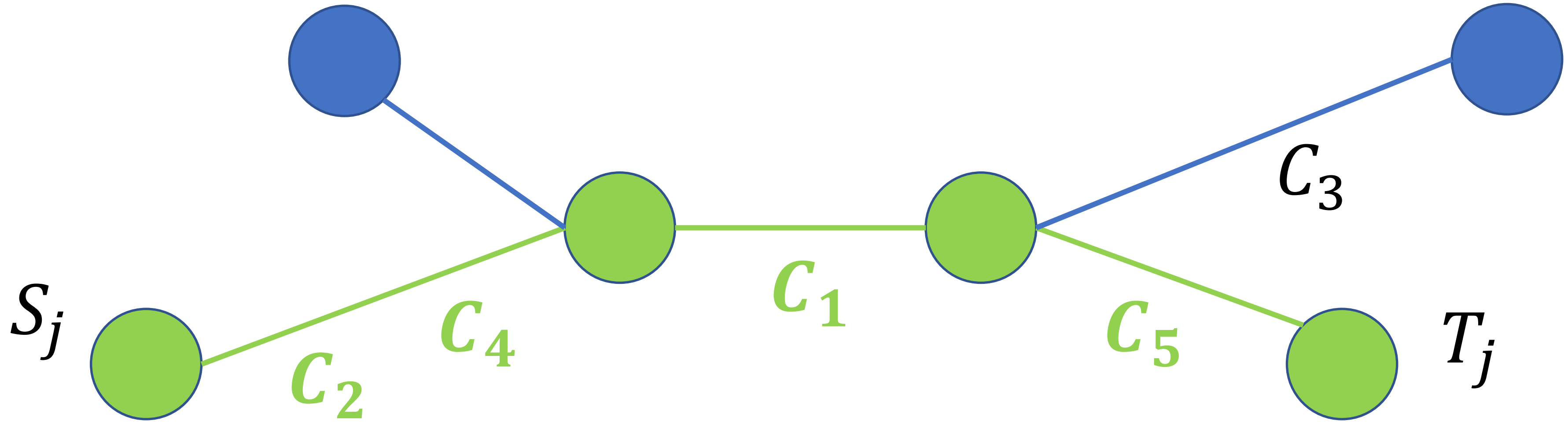
小課題３（パス）,満点解法



- （C_iが小さいものから最大何個の和がY_j以下か）を求める
- **パスS_j-T_jにあるC_iであって、全てのC_iを昇順にソートしたときk番目までに入るものの和がY_j以下**となる最大のkは何か？
- このままでは難しい

→二分探索したときの**判定問題**を考える

小課題３（パス）,満点解法



- **パス$S_j - T_j$にあるC_iであって、全てのC_iを昇順にソートしたときk番目までに入るものの和がY_j以下か？**
- このQ個の判定問題を解く
- 何もない状態から始めて、C_iを昇順に辺上に追加していく
- k番目まで追加したら、パス$S_j - T_j$上の和を求めてY_j以下か判定

小課題３（パス）,満点解法

パスの場合

- 一点更新
- 区間の値の和の取得

→**BIT**

判定問題Q個が$O((M+Q)\log N)$で解ける

小課題３（パス）,満点解法

木の場合

- 一辺の重みを更新
- $S_j - T_j$パスの重みの和の取得

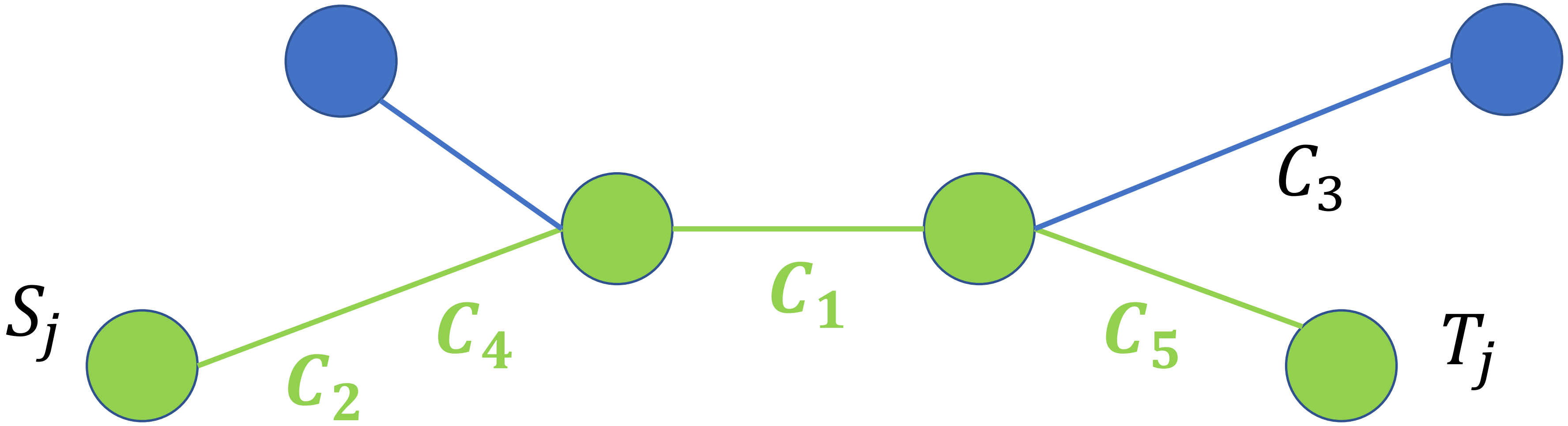
→**LCA＋EulerTour＋BIT**

判定問題Q個が$O((M+Q)\log N)$で解ける

小課題３（パス）,満点解法

- 二分探索したときの判定問題Q個が$O((M+Q)\log N)$で解けた
- これを$O(\log M)$回繰り返して二分探索を行えば、元の問題Q個が$O((M+Q)\log N\log M)$で解ける
- **並列二分探索 (Parallel Binary Search)**と呼ばれる

小課題３（パス）,満点解法



- パス$S_j - T_j$上にあるC_iであって、全てのC_iを昇順にソートしたときk番目までに入るものの和がY_j以下となる最大のkが分かった
- (C_iが小さいものから最大何個の和がY_j以下か）を求める
- k番目まで追加した後、パス$S_j - T_j$上にあるC_iの個数を求める
- 同様の手法でよい → $O((M+Q)\log N \log M + N \log N + M \log M)$

余談

- 並列二分探索をせず、Mo’s algorithmを使う方法もある
- （簡単のため$N = M = Q$とする）
- 更新　$O(N^{1.5})$回
- 計算　$O(N)$回
- BITだと両方に$\log N$が付いて$O(N^{1.5} \log N)$
- 平方分割で更新$O(1)$, 計算$O(\sqrt{N})$にすると$O(N^{1.5})$

得点分布