

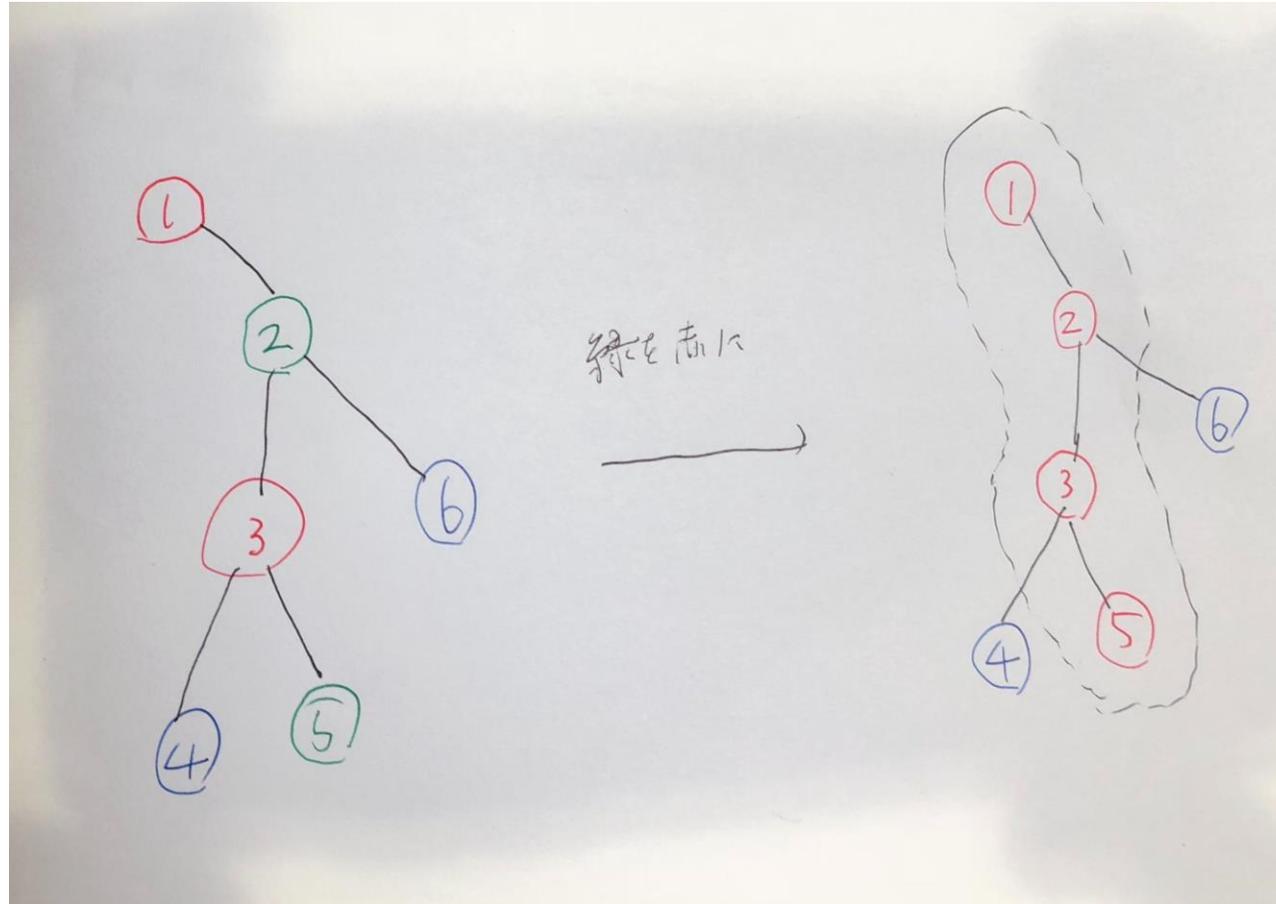
Capital City

伊佐 碩恭 (HIR180)

問題概要

- 木が与えられ，各頂点は何らかの都市に属している
- 都市と都市の合併を繰り返すことにより，「その都市に含まれるどの2頂点間も，その都市に含まれる頂点のみを通過することで行き来できるような都市」を作りたい
- 最小で何回合併を行えば良いですか？

入力例1



言い換え

- グラフ理論の通称に従って，町を頂点と言い換えます
- この問題において，任意の連結な部分木が首都として指定できるので，首都を連結な部分木と同義のものとして解説します

小課題1

- $N \leq 20$
- 「最終的に合併されて1つの都市になる」ような都市の集合を全て探索することができる
- 集合が決まれば、いずれかの都市に含まれる頂点たちが全域木を成すかどうかで判定できる
- $O(2^N * N)$
- 1点

小課題2

- $N \leq 2000$
- 最終的に首都に含まれる頂点を1つ決め打つと，首都に必ず入らなくてはならない都市の集合が定まる（決め打った頂点を v とすると， v と同じ色の頂点と， v を結ぶパス上にある頂点の都市もまた最終的に首都に含まれる… という操作を新たに都市が追加されなくなるまで繰り返す）
- 重要な性質として，この操作によりできた首都は頂点 v を含む首都の中で**極小**であることが容易にわかる

小課題2

- この性質より，各頂点を始点として前ページの処理を行い最小値を取れば答えが求まる
- $O(N^2)$
- 10点

小課題3

- 木は直線グラフ
- 色々な方針があります
 1. 数列の問題だと思おう
 2. 分割統治
 3. Segment tree状に辺を貼る

小課題3 – 方針1

- 左端Lを決め打ちすると、右端Rに対する条件は以下のようなになる:
 - $[L,R]$ に含まれる要素について、「自分と同じ都市の頂点で、左方向に一番近い頂点番号」が存在しないか、すべてL以上
 - $[L,R]$ に含まれる要素について、「自分と同じ都市の頂点で、右方向に一番近い頂点番号」が存在しないか、すべてR以内
- これは、segtreeやstd::setを使うことで実装できる
- 30点

小課題3 - 方針2

- 今、頂点 $[l,r]$ に対して考えているとする
- 頂点 $m = (l+r)/2$ を含む場合と含まない場合に分ける
- 含む場合は、小課題2の方針で $O(r-l)$ で解ける
- 含まない場合は、 $[l,m-1]$ $[m+1,r]$ に分割して同じ処理をする
- 困るのは今見ている範囲外の町を含めないといけない場合
 - 実は、そのような場合は処理をやめて良い (今までに求めた首都のいずれかを包含する -> その時に求めた首都を包含する)

小課題3 – 方針3

- 都市 c を首都に含めることにすると，都市 c に含まれる町のうち一番左にあるものを L_c ，一番右にあるものを R_c とすると， $[L_c, R_c]$ の町は全て首都に含める必要が生じる
- これを素直なグラフの問題に落とすと，都市 c に対応する頂点から町 L_c, \dots, R_c に対応する頂点へ辺を張り，できたグラフを強連結成分分解して，出次数0の強連結成分に対して含まれる都市に対応する頂点の個数の最小値 - 1 を求める問題になる

小課題3 – 方針3

- このままでは辺数が $O(N^2)$ なので正答できない
- が、区間に辺を張る という特性を使いオーダーが落とせる
- Segment tree (葉ノードが1つの町に対応する頂点) に対し、親から子へ有向辺を張ったグラフを考えると、 $[l,r]$ に辺を張る という処理は $O(\log N)$ 個の区間に対して辺を張れば良くなる
- よって、強連結成分分解は頂点数と辺数の和に線形な時間で求まるため、全体で $O(N \log N)$ 時間の解法になる
- 類題: https://atcoder.jp/contests/arc069/tasks/arc069_d

満点解法

- こちらも色々な方針があります
 1. 重心分解 (<- 小課題3の方針2の拡張)
 2. 強連結成分分解 (<- 小課題3の方針3の拡張)

満点解法 - 方針1

- 小課題2の性質を思い出す
 - 「最終的に首都に含まれる頂点を1つ決め打つと，首都に必ず入らなくてはならない都市の集合が定まる」
- 頂点を決め打つ時に性質の良い頂点を選ぶと上手くいく??
- **重心分解**

満点解法 - 方針1

- 木の重心を使うと決め打てば、今の部分木サイズに線形な計算量で最小値が求まる
- 木の重心を使わないケースは、重心分解して分かれた連結成分ごとに考えれば良い
- 今見ている部分木の外に頂点が存在する都市を使わないと行けなくなった場合は、計算を中止して良い (今までの重心のいずれかは使わないといけなくなる \rightarrow その場合の首都を包含するため、このケースは考えなくて良い)
- $O(N \log N)$ 時間で解ける

満点解法 - 方針2

- 小課題3の時は区間に辺を張る だったので簡単だったが、今回は木のパス上に辺を張る必要がある
- 木を直線グラフに分解する方法があったような…?
 - Heavy Light Decomposition
- $O(\log N)$ 個の直線グラフに対し、区間に辺を張る 問題に変換できた
- 先ほどと同様にして、 $O(N \log^2 N)$ 時間で解ける
 - 実装によってはTLEするかも

満点解法? - 乱択アルゴリズム

- 小課題2と同様にして，最終的に首都に含まれる頂点 v を決め打って極小な首都を求めることを考える
- 新たに都市 c を追加しないといけなくなった場合，**都市 c に含まれる頂点を使うと決め打って計算したことがある**ならば処理を打ち切って良い（その場合の首都を包含するから）
- 使うと決め打つ頂点の順番や，首都を求める際に次に追加すべき都市を見つける際の処理をランダムに行うことによって，非常に高速に動作した
- 解析は簡単ではない
 - 少なくとも期待値で $O(N \log N)$ 時間かかるケースは存在する