Problem J: Tree Allocation

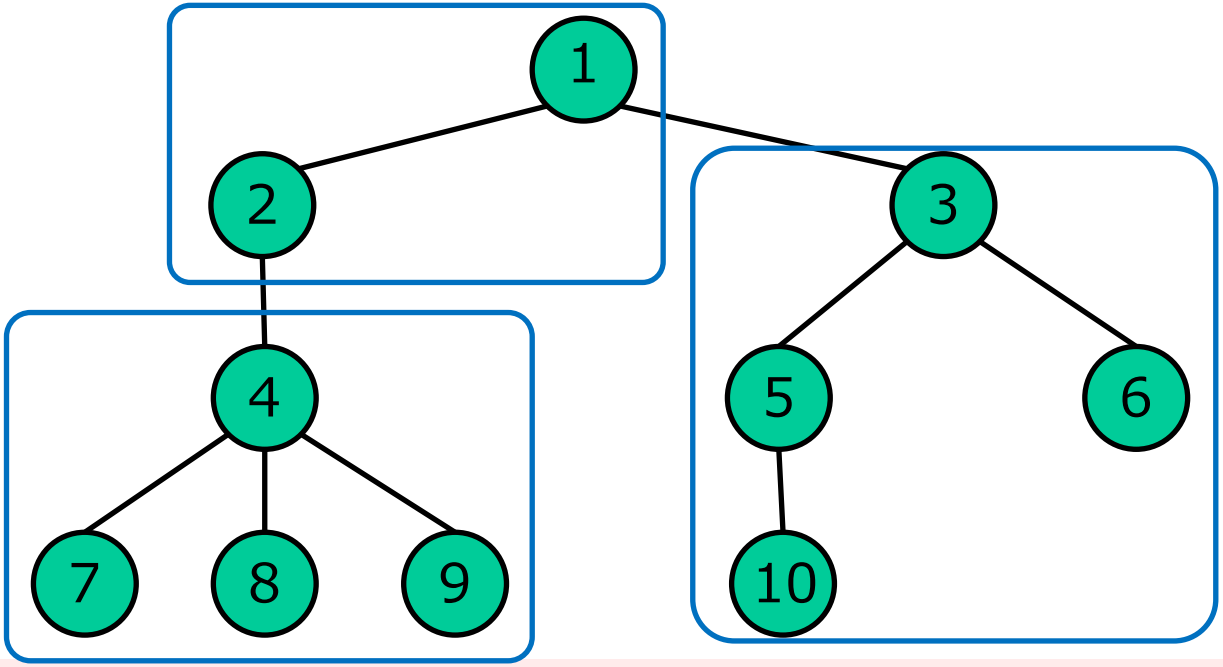
原案：吉田
解答：山口，須藤
解説：須藤

問題概要

- 木の頂点を高々B個ずつのグループに分ける
  - 同じブロックに属する頂点間の移動はコスト0
  - 異なるブロック間の移動はコスト1
  - さらに初期コストが1必要
- 木の各頂点を根とした場合について，葉までのパスをたどるときのコストのうち最大のものを最小化
- 木の頂点数 $N \leqq 10^5$, ブロックサイズ $B \leqq N$

例(Sample Input 3)

- N＝10, B=4, 根を頂点1としたとき
  - 以下の割当てで，すべての葉へコスト2以内で行ける
  - コスト1以下の割当ては存在しないので，最小コストは2



解法 Step1 (1/4)

- 初めに根を固定したときの最小コストについて考える

- 各頂点vについて以下の2種類の値を持っておく
  - depth：子ノードを経由して行けるすべて葉のうち，
    コストが最大のものを最小化したときの値
  - size　：コストdepthを達成するときの，
    頂点vを含むブロックの最小の大きさ

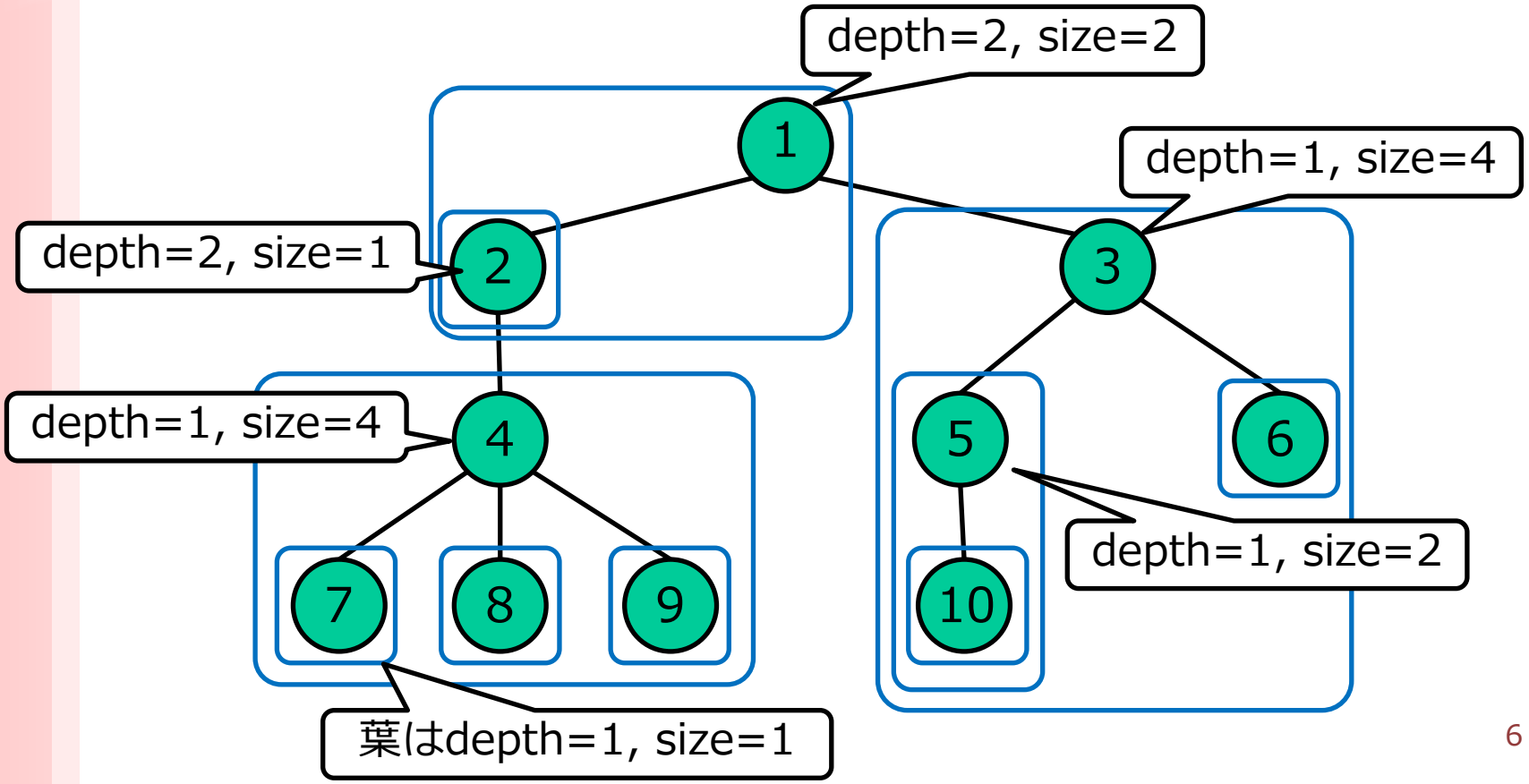
- 頂点1を根とした木について，葉からこの値を調べる
  - 頂点1を根とした時のトポロジカル順序は自明なので，実装も楽

解法 Step1 (2/4)

- 葉となる頂点
  - 子ノードがないので，depth = 1, size = 1
- それ以外の頂点
  - 子ノードのdepthの最大値をDとする
  - depthがDである子ノード以外は考えなくて良い
    - ブロックに含めてもコストの最大値は良くならない
  - depth=D となる子ノードのsizeの和Sについて，
    - S+1≦B のとき，現在の頂点と深さ最大の子ノードを同じブロックにまとめると，現在の頂点からdepth=Dが達成可能
    - S+1>B のときは depth=D にできないので，depth = D+1, size=1とする

解法 Step1 (3/4)

- 例(Sample Input 3：N=10, B=4)

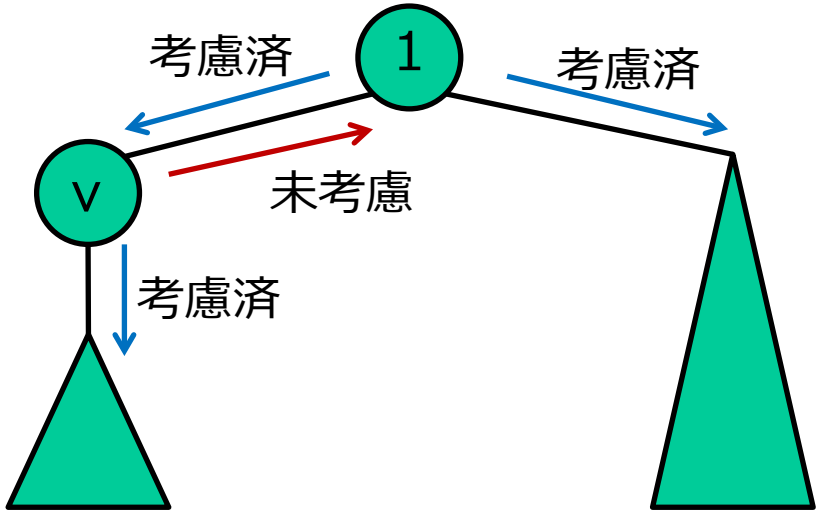


解法 Step1 (4/4)

- 頂点1のdepthの値が，頂点1を根としたときの最小コストとなる
  - 前スライドの例では，最小コストが2となる
- ここまでの計算量はO(N)
  - Nが最大で10^5なので，この方法をN回実行するとTLE
  - 他の頂点について，より効率的にコストを求める必要がある

解法 Step2 (1/3)

- 頂点1に隣接する頂点vに注目する
  - 頂点1について最小コストを調べた段階で，
    - 頂点1から見て，depthとsizeはすべての子を考慮した値
    - 頂点vから見て，depthとsizeは頂点1以外の子を考慮した値
  - 頂点1について，頂点vを考慮しない場合のdepthとsizeを求められれば，頂点vを根としたときの最小コストが求められる



解法 Step2 (2/3)

- 頂点1から，頂点vを見ないときのdepthとsize
  - 頂点1が，頂点vよりもdepthの値が大きい子を持つとき
    - 元から頂点vは考慮されてないのでdepthとsizeは不変
  - それ以外(頂点vは最大のdepthをとる子ノードの1つ)
    - 頂点vと等しいdepthをとる(頂点1の)子ノードが存在すれば，depthが最大となる頂点v以外の子ノードを考慮した時の値
    - そうでなければ，depthが2番目に大きい子ノードを考慮した時の値

- 上記を効率良く求めるため，以下の値を保持しておく
  - (見た子ノードのうち)depthが最大となる子ノードのdepthと，depthが最大となる子ノードのsizeの和
  - depthが2番目に大きい子ノードについて，同様の値

解法 Step2 (3/3)

- 最小コストを求めた頂点に隣接する頂点に関して，この方法で最小コストを求めることが出来る
  - メモ化しておくことにより，1頂点につきO(1)
  - 最初に最小コストを求めた頂点から，幅優先で最小コストを求めていけば全体でO(N)
- Step1と合わせて，計算量は全体でO(N)

備考

- Step1をDFSでやるとスタックオーバーフローの危険性
  - 頂点が直線状に並んだテストケースが入っています
- (頂点, 親)の組み合わせでのメモ化はTLE
  - 次数の多い頂点が存在すると計算量が$O(N^2)$になってしまう
  - これを落とす用のテストケースも用意していましたが，活躍の場はありませんでした
- 入力，出力のサイズが大きめ
  - コンテストでは，遅い入出力を使っても間に合うようにTime Limitを設定していました(10sec.)
  - C++ならcinではなくscanfなど，早い入出力だと安心

ジャッジ解

- 山口　128行(3,145B), Java
- 須藤　102行(2,593B), C++

結果

- Submitチーム数：1
- Acceptチーム数：1
- 総Submit：2

- First Accept：mlTwTlm (4h15m)