2011/05/14 東京大学駒場キャンパス

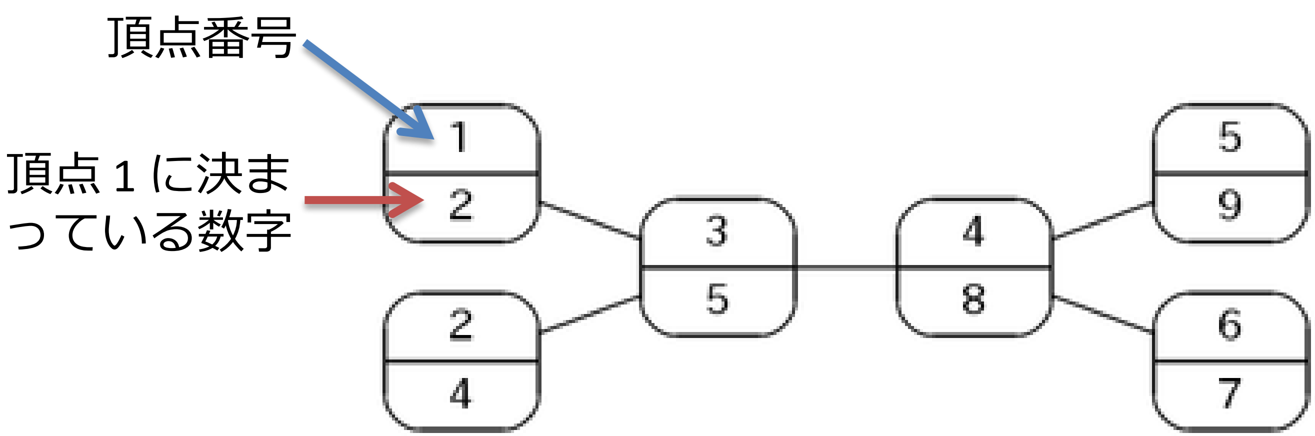
東京大学プログラミングコンテスト 2011
問題 L : L 番目の数字

東京大学大学院情報理工学系研究科
秋葉 拓哉

問題概要

- 各ノードに数字が決まってるツリーがある
- 以下のタイプのクエリを大量に処理:

ある頂点 v から w への経路上の数字で，
l 番目に小さい物を求めよ

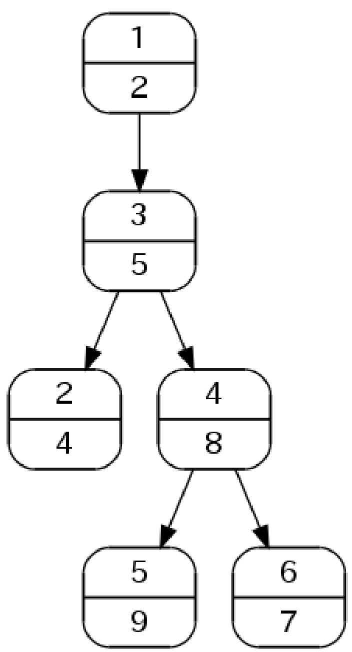


例: 頂点 1 から頂点 6 への経路での，3 番目
→ 2, 5, 8, 7 の 3 番目 → 7 が答え

問題を変形

- 頂点 1 を根にして，向き付きの木にする
- 以下を効率的に計算できるか？

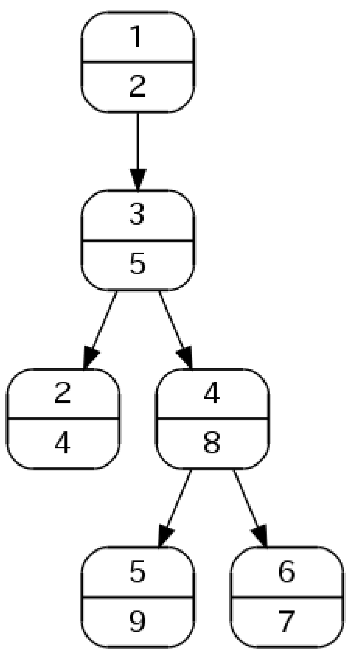
頂点 v から根までで x 以下が何個あるか？



例: 頂点 4 から根に 6 以下は何個あるか？
→ 8, 5, 2 に 6 以下は何個あるか？
→ 2 個

それができると？

- それができると，**任意の 2 頂点間の x 以下の数字の個数**が求められる
  - 引き算すればよい

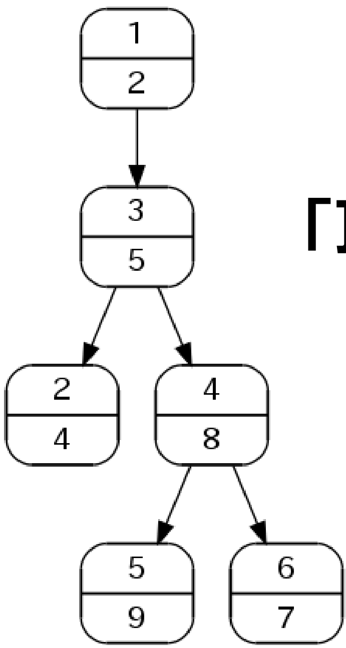


例:
頂点 2 から頂点 4 の x 以下の個数
= (頂点 2 から上) + (頂点 4 から上)
 - (頂点 3 から上) + (頂点 1 から上)

配列におけるスライド和と同様のアイディア

で，じゃあそれができると？

- x 以下の個数がちょうど k 個以上になる x を**二分探索**で探せば OK



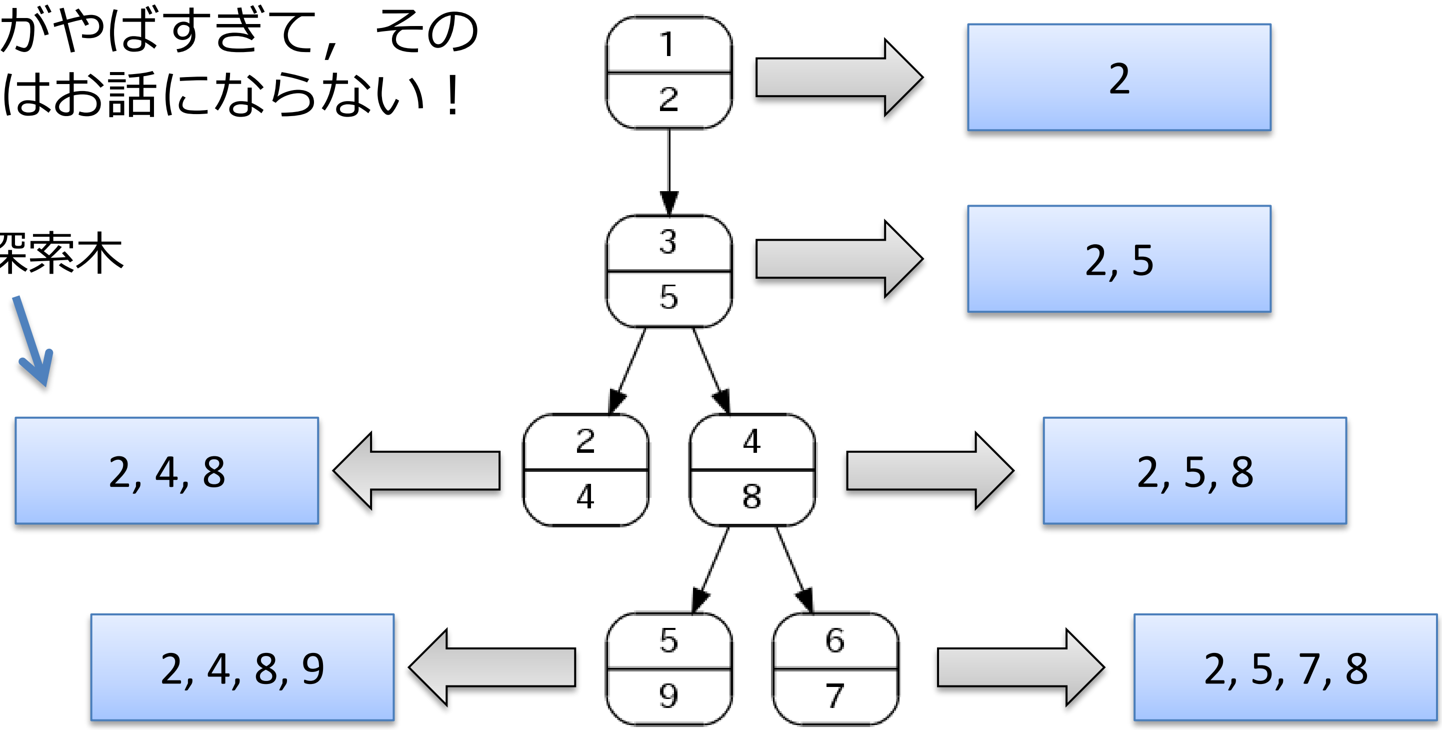
では，最初の
「頂点 v から根までで x 以下が何個あるか？」
はどうやって計算すれば？

一見，頭悪そうな方法

- 各ノードが，二分探索木を持つ
- そのノードから根までの数字を全部突っ込む

メモリがやばすぎて，その
ままではお話にならない！

二分探索木



しかし…

- 小ノードの二分探索木と親ノードの二分探索木は，あまり差がない
  – 具体的には，1 個数字が追加されるだけ

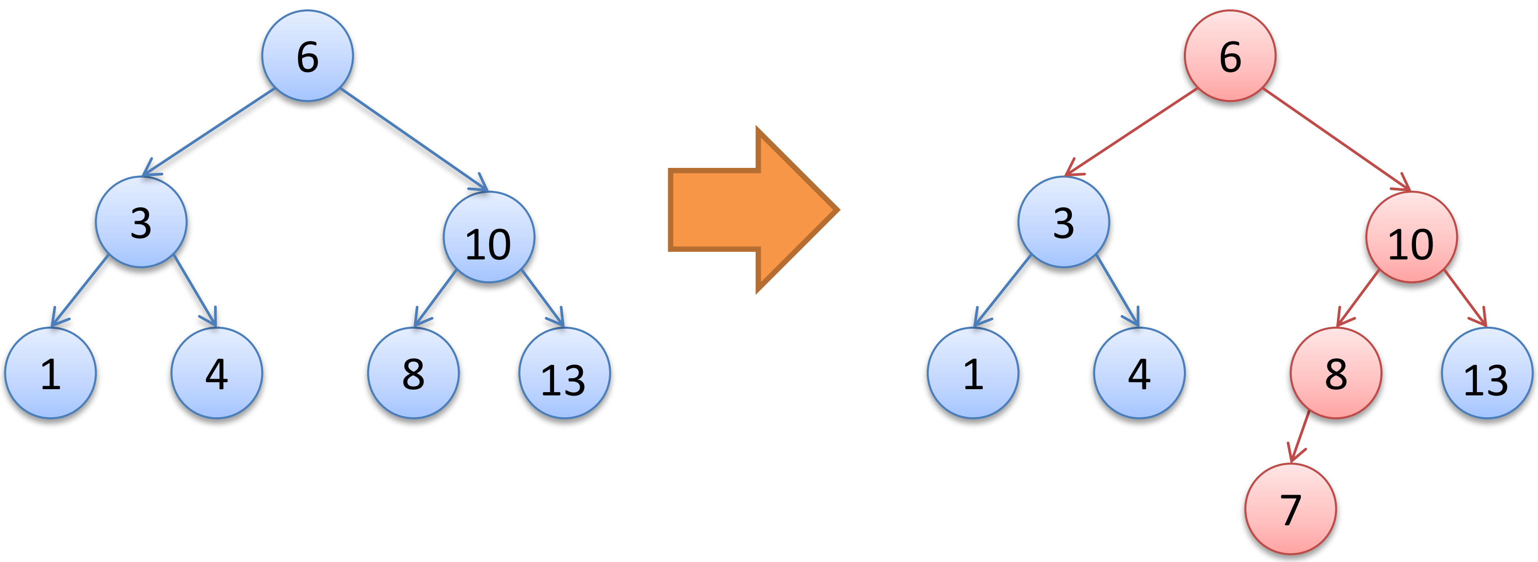
- **親ノードの木の大部分を再利用**した木を構成すれば良い！

どういう事？

親の二分探索木

子の二分探索木

子ノードの数字は 7 とすると，
7 を親の二分探索木に追加



赤のノードを新しく作って，青のノードは使い回す！
高さ (= 平衡されてれば $O(\log n)$) の分だけ領域が増えるだけ

“永続データ構造”

- こういうのを，**永続データ構造**と言います
  - 操作の度に書き換えるのが普通だが，書き換えない
  - 操作をすると，新しいデータ構造が帰ってくる
  - 操作をする前の，昔のやつも生き続ける

- 関数型言語でおなじみ

それを作れば

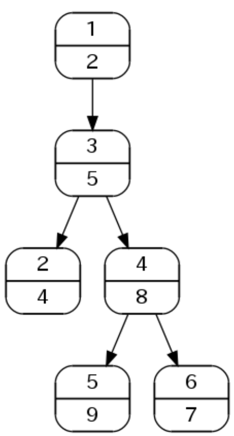
- そのまま実装すると
  - 領域 $O(n \log n)$，時間 $O(n \log n + q \log^2 n)$
  - 二分探索木は平衡処理が必要で面倒…

- そこで，二分探索木の形を最初に固定すると良い
  - つまり，根ノードの段階で，個数 0 として全ての数字を挿入
  - すると，全てのノードが持つ二分探索木の形が同じになる
  - 二分探索が，二分探索木を巡りながら行えるようになり，計算量も落ちる
  - 実装も楽になる

領域 $O(n \log n)$，時間 $O\big((n + q) \log n\big)$

別解

- Wavelet Tree (もどき) を拡張する
  - 領域効率はどうでも良いので，ビットベクトルを用いた簡潔な表現をしなくて良いという意味で「もどき」
- 2 つの拡張
  - 各アルファベットが，出現するだけじゃなく，減りもする
  - 1 つの区間でなく，2 つの区間に関するクエリを処理する



Euler-Tour

頂点	1↓	3↓	2↓	2↑	4↓	5↓	5↑	6↓	6↑	4↑	3↑	1↑
列	+2	+5	+4	-4	+8	+9	-9	+7	-7	-8	-5	-2

領域 $O(n \log n)$，時間 $O((n + q) \log n)$

提出状況

- 正解: なし
- 提出: 17 件