Construction of Highway

高谷悠太

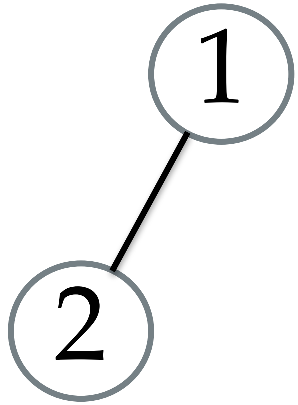
問題概要

- 木の頂点を根に近い順に追加していく。
- 木の頂点には活気という値があり、頂点を追加するたびに、活気の列の inversion を求める。
- そして、活気の値を書き換える。

サンプル

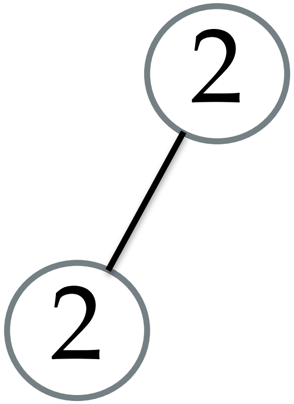
①

サンプル

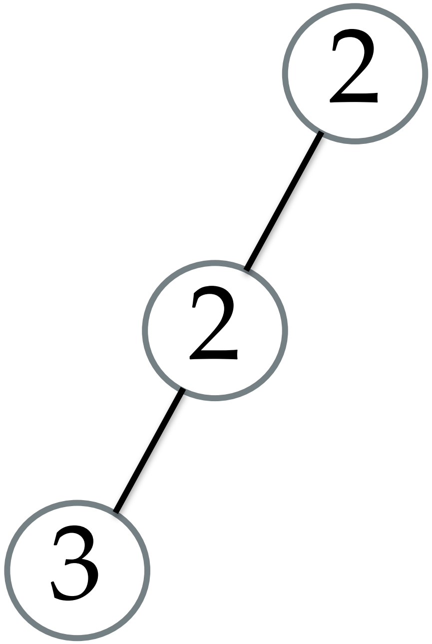


1なので、
inversionは0

サンプル

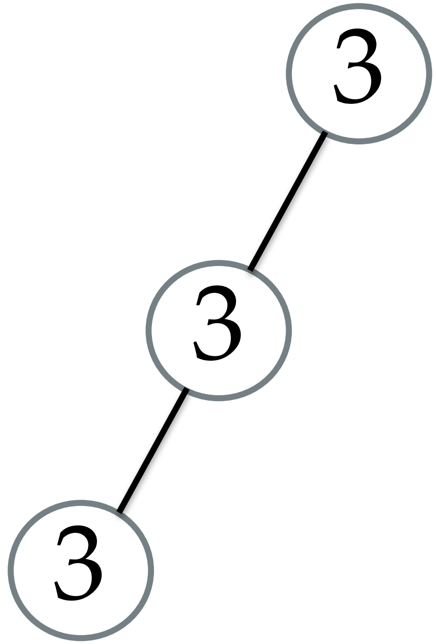


サンプル

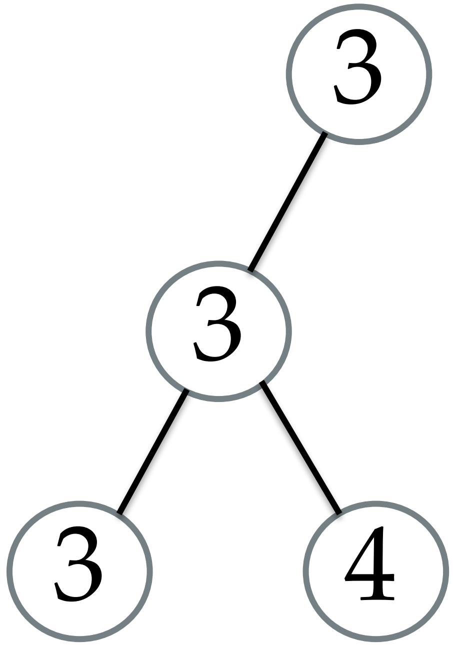


2,2 なので、
inversion は 0

サンプル

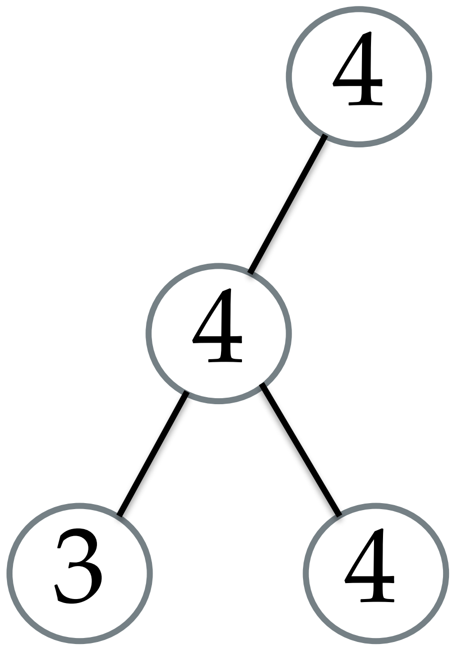


サンプル

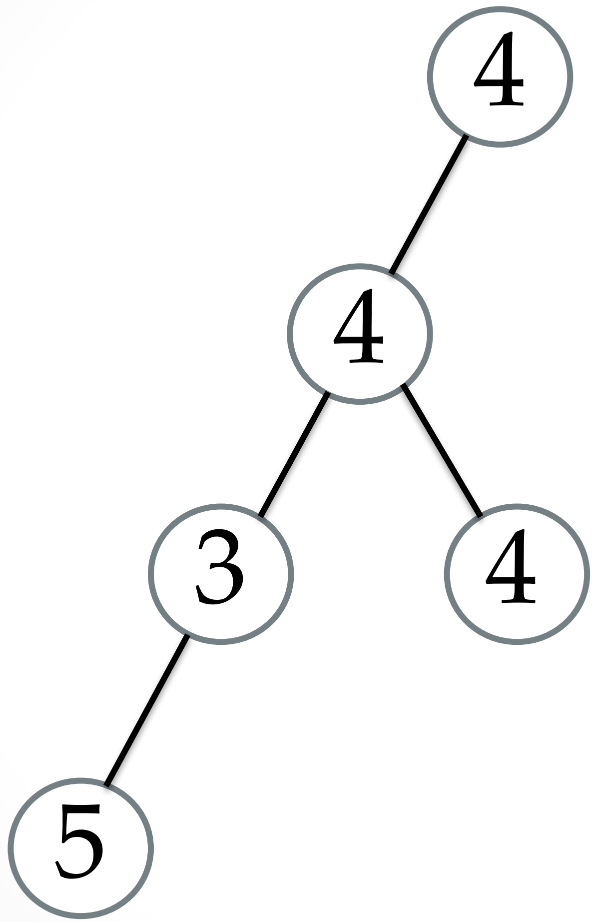


3,3 なので、
inversion は 0

サンプル

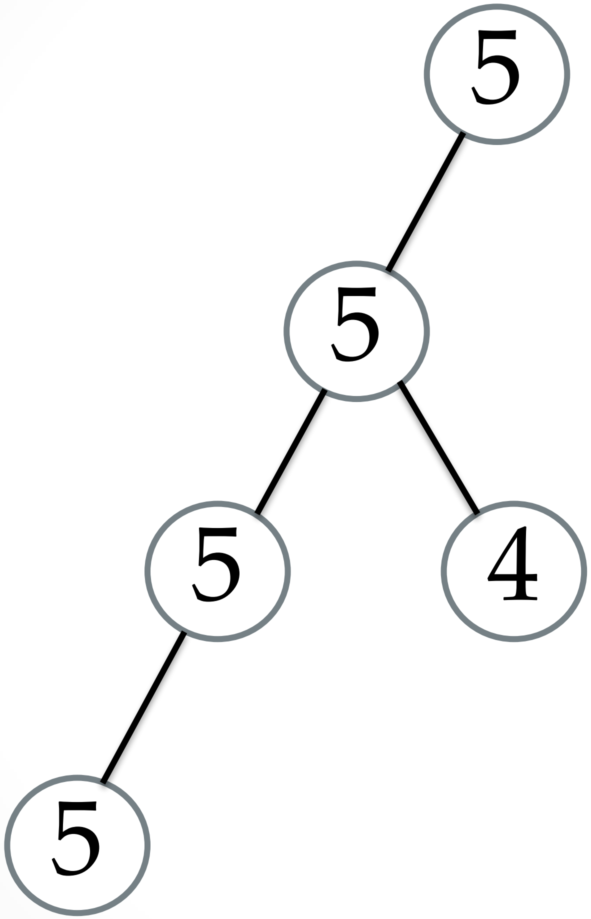


サンプル



4,4,3 なので、
inversion は 2

サンプル



制約

- 小課題1 : $1 \leqq N \leqq 500$
- 小課題2 : $1 \leqq N \leqq 4000$
- 小課題3 : $1 \leqq N \leqq 100000$

制約

- 小課題1 : $1 \leqq N \leqq 500$ $\rightarrow$ $O(N^3)$
- 小課題2 : $1 \leqq N \leqq 4000$ $\rightarrow$ $O(N^2)$
- 小課題3 : $1 \leqq N \leqq 100000$ $\rightarrow$ $O(N * \log N)$
- ぐらい？？

小課題1

- $N-1$ 回それぞれで $O(N^2)$ かけてよいから、根までのパスを調べることができる

- 更新も簡単

小課題1

- $N-1$ 回それぞれで $O(N^2)$ かけてよいから、根までのパスを調べることができる

- 更新も簡単

- inversion の計算を $O(N^2)$ でできればよい

小課題1

- $N-1$ 回それぞれで $O(N^2)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(N^2)$ でできればよい
- 愚直で大丈夫

小課題1

- $N-1$ 回それぞれで $O(N^2)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(N^2)$ でできればよい
- 愚直で大丈夫

小課題2

- $N-1$ 回それぞれで $O(N)$ かけてよいから、根までのパスを調べることができる

- 更新も簡単

小課題2

- $N-1$ 回それぞれで $O(N)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(N)$ ぐらいでできればよい

小課題2

- $N-1$ 回それぞれで $O(N)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(N)$ ぐらいでできればよい
- BIT で大丈夫

満点解

- $N-1$ 回それぞれで $O(1)$ かけてよいから、根までのパスを調べることができる

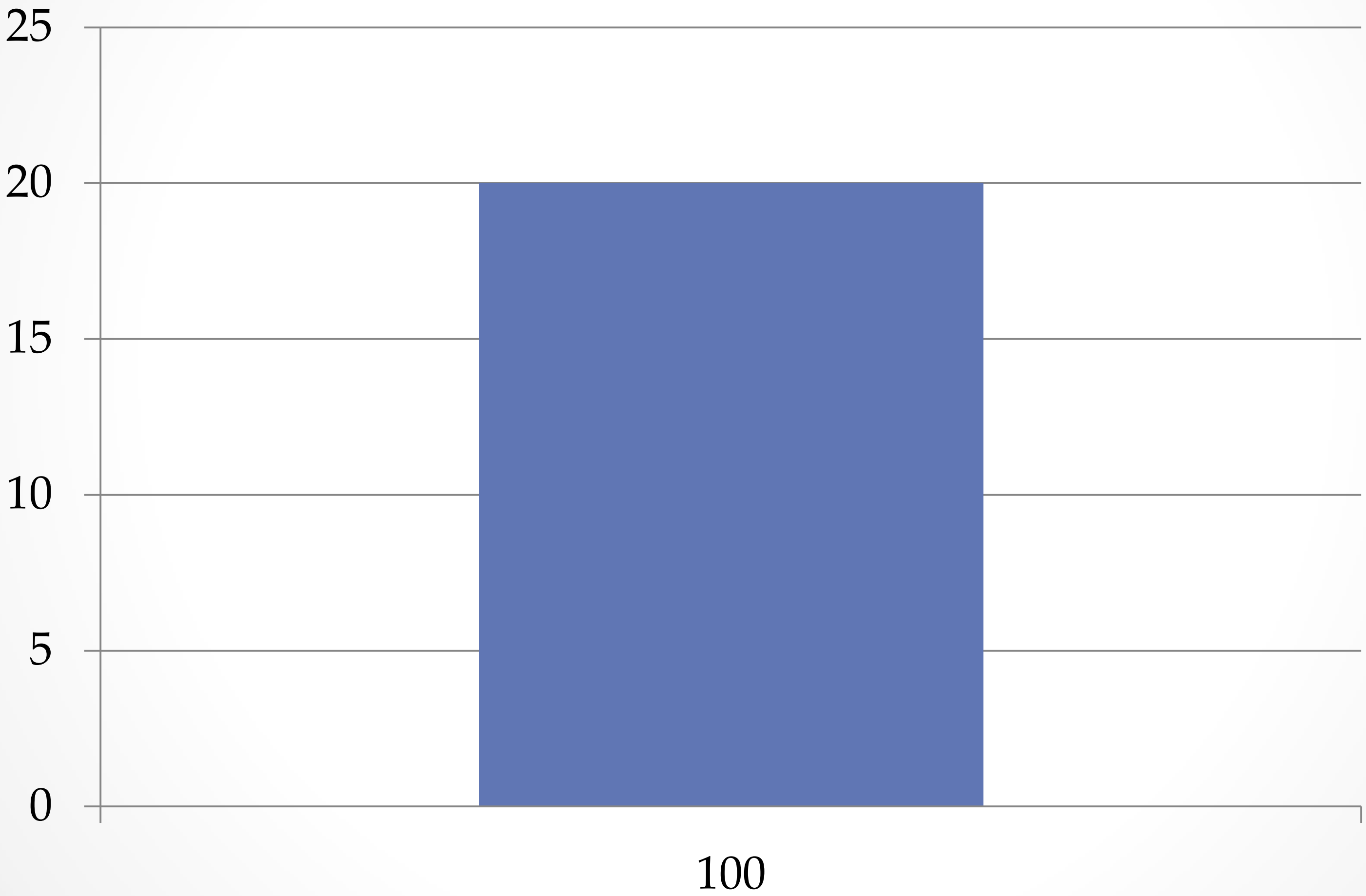
- 更新も簡単

- inversion の計算を $O(1)$ でできればよい

満点解

- $N-1$ 回それぞれで $O(1)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(1)$ でできればよい

得点分布



満点解

- $N-1$ 回それぞれで $O(1)$ かけてよいから、根までのパスを調べることができる
- 更新も簡単
- inversion の計算を $O(1)$ でできればよい
- ？？？？？？

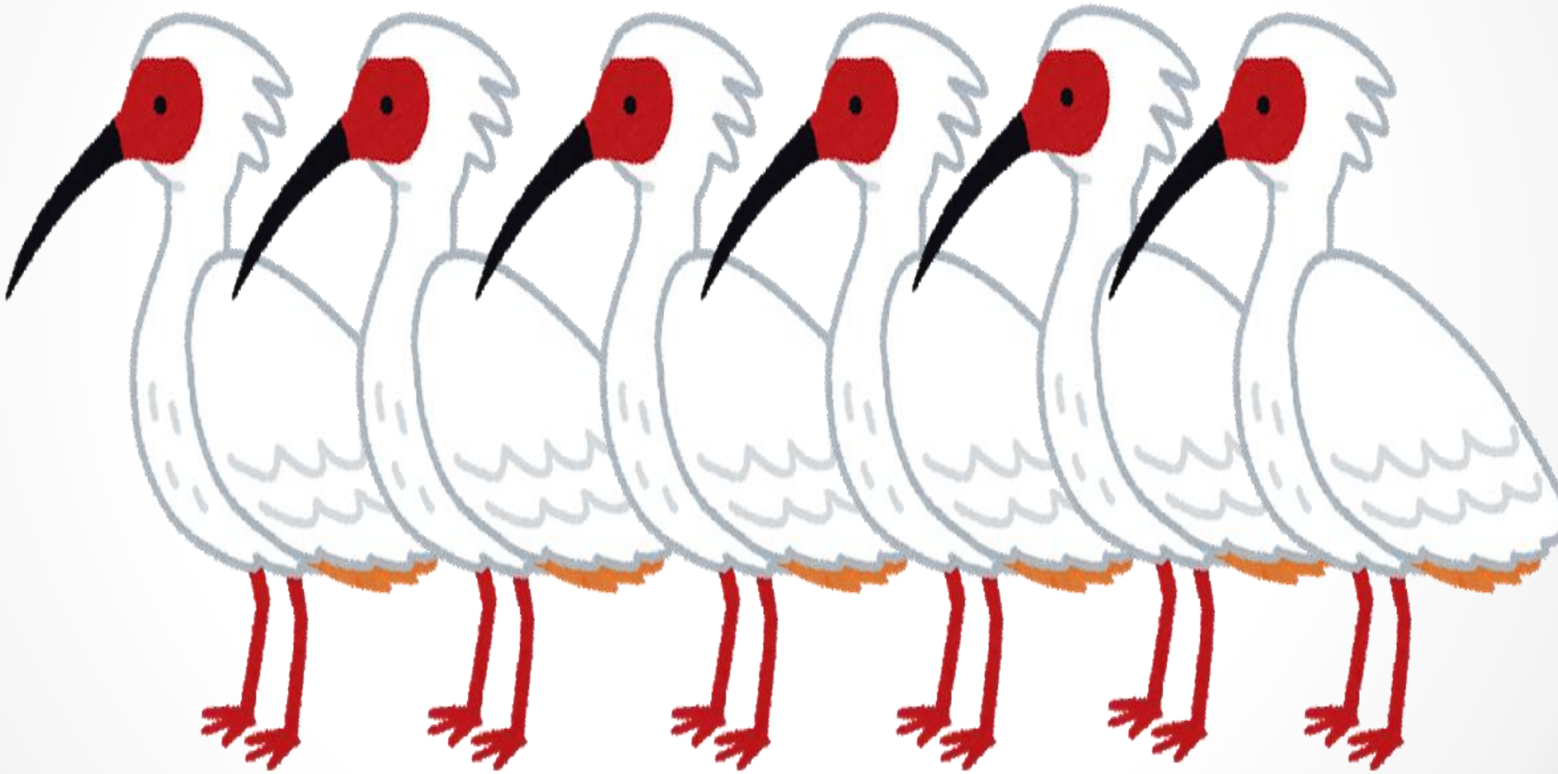
よく考えよう

- 今までの小課題では全く問題の性質を使っていない
- N 回ぐらい inversion の計算をしてるだけ

よく考えよう

- 今までの小課題では全く問題の性質を使っていない
- N 回ぐらい inversion の計算をしてるだけ
- 具体的な例で考えてみよう！！

列のトキ



列のとき

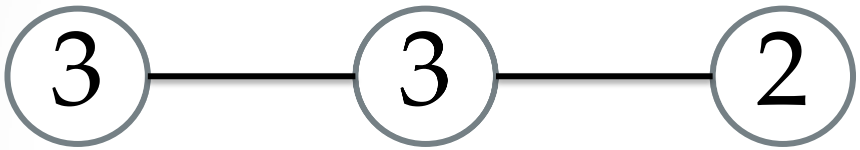
④

列のとき

4 なので、
inversion は 0

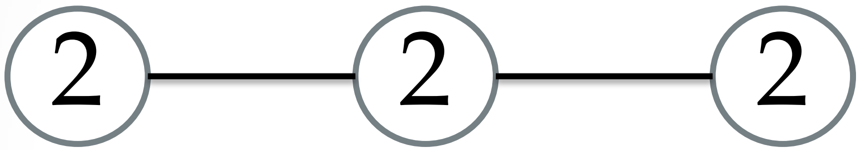
列のとき

列のとき



3,3 なので、
inversion は 0

列のとき



列のとき

2,2,2 なので、
inversion は 0

列のとき

列のとき

列のとき（拡張）

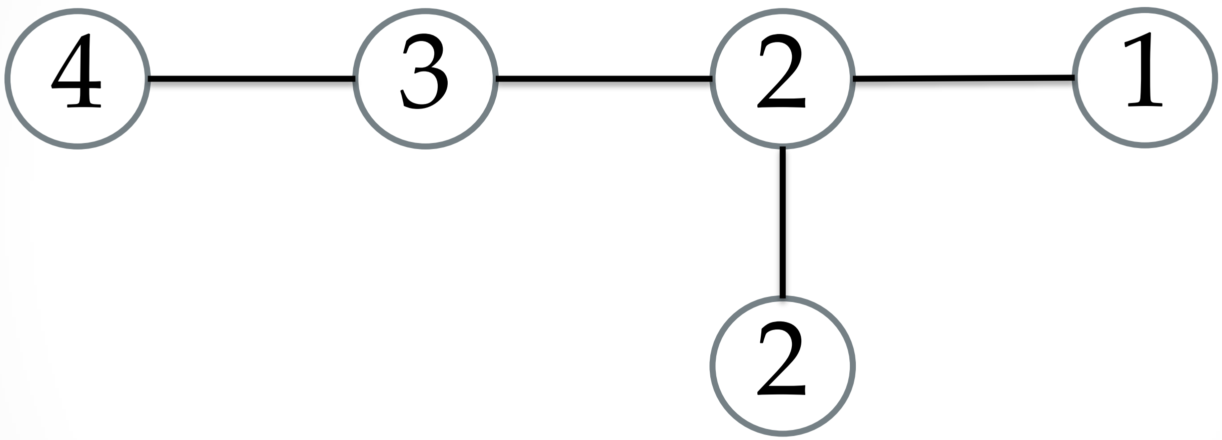
- 列を作るのは自明なケースに帰着されてしまう

列のとき（拡張）

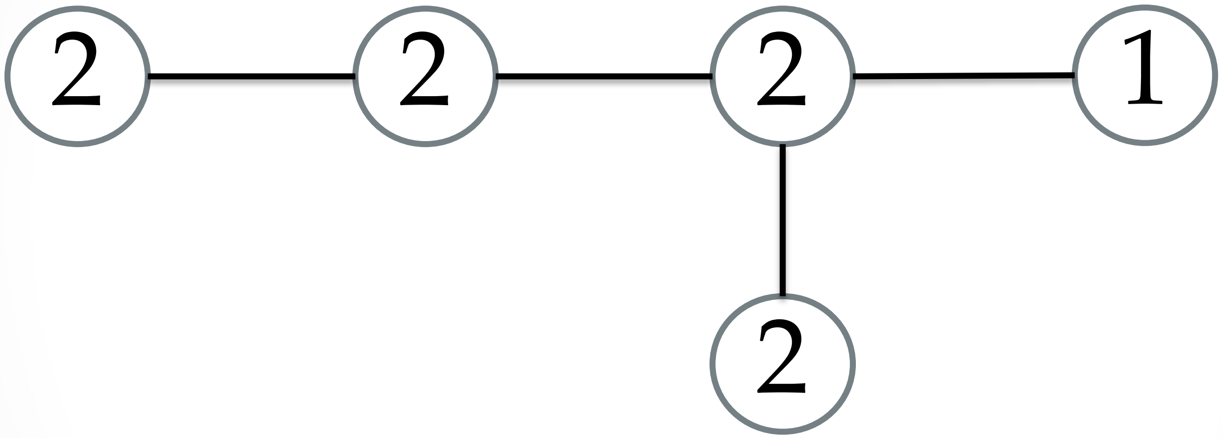
- 列を作るのは自明なケースに帰着されてしまう
- 列がある状態からその列を更新するのは非自明っぽい

列のとき（拡張）

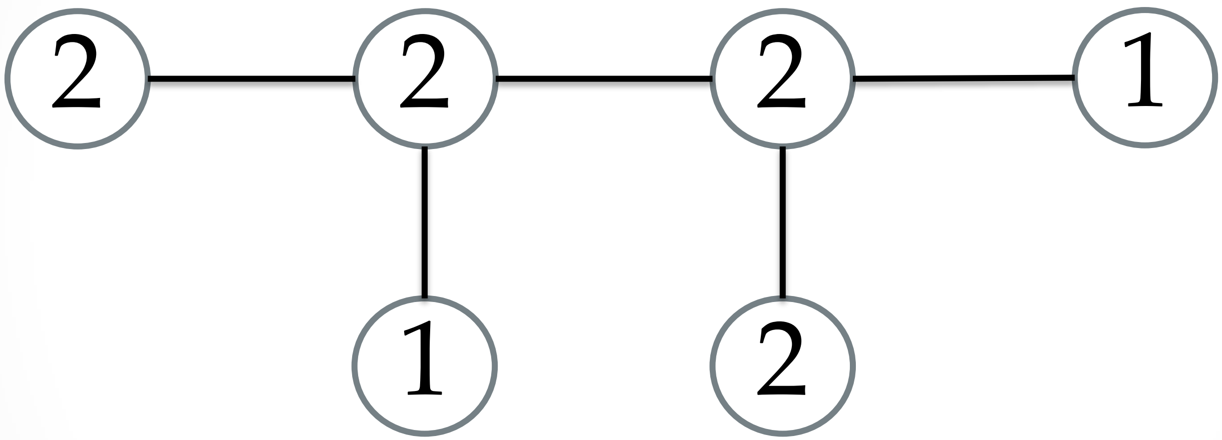
列のとき（拡張）



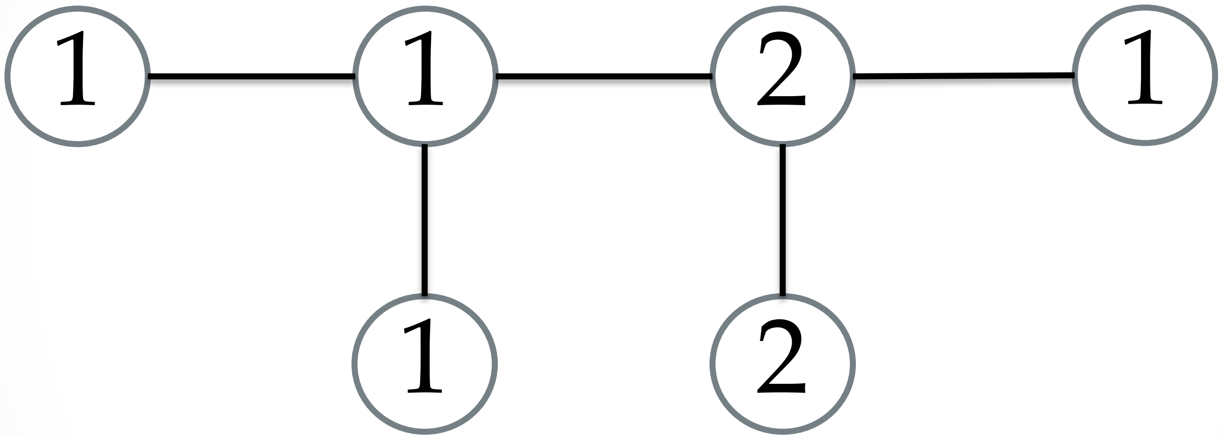
列のとき（拡張）



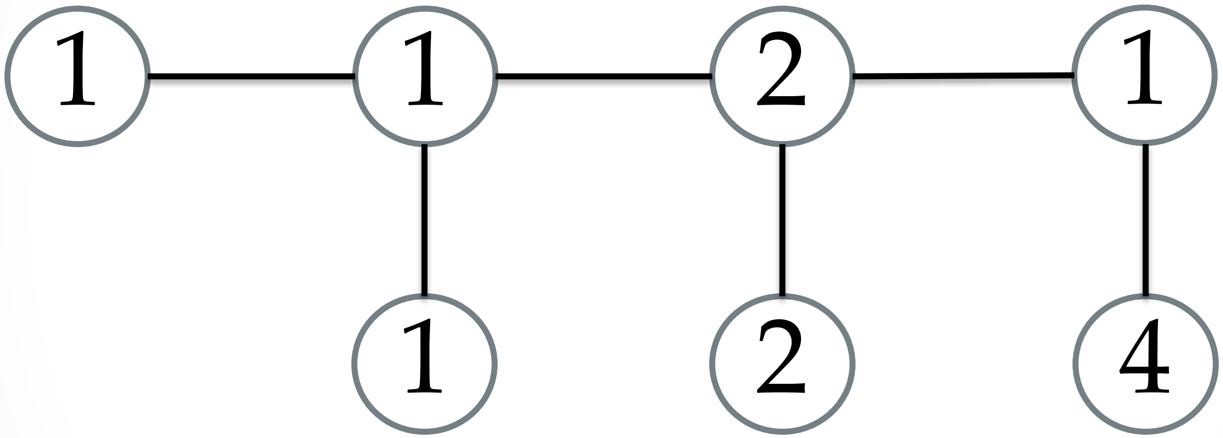
列のとき（拡張）



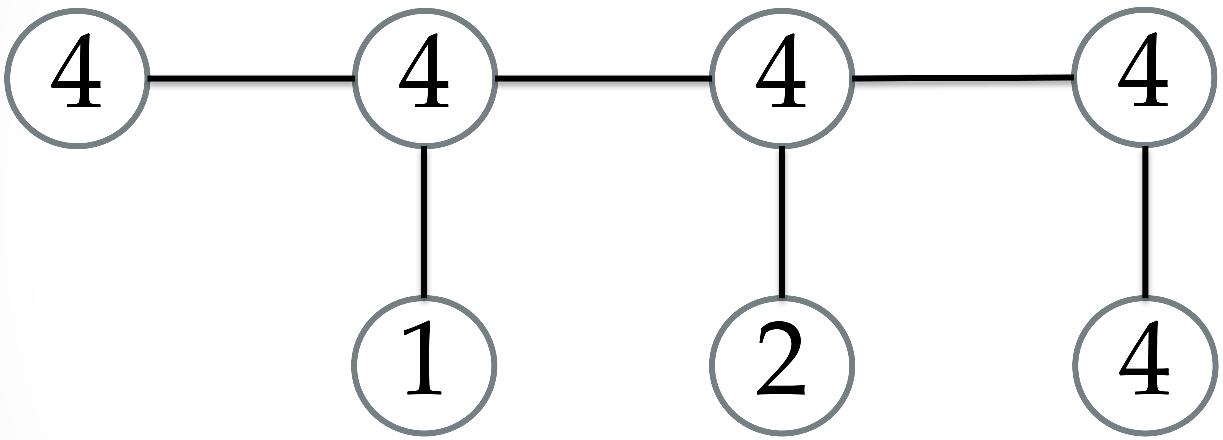
列のとき（拡張）



列のとき（拡張）



列のとき（拡張）



列のとき（拡張）

- 他の部分を無視して、今考えてる列にだけ注目する
- 先頭 k 文字を x にする、という変更が起こる

列のとき（拡張）

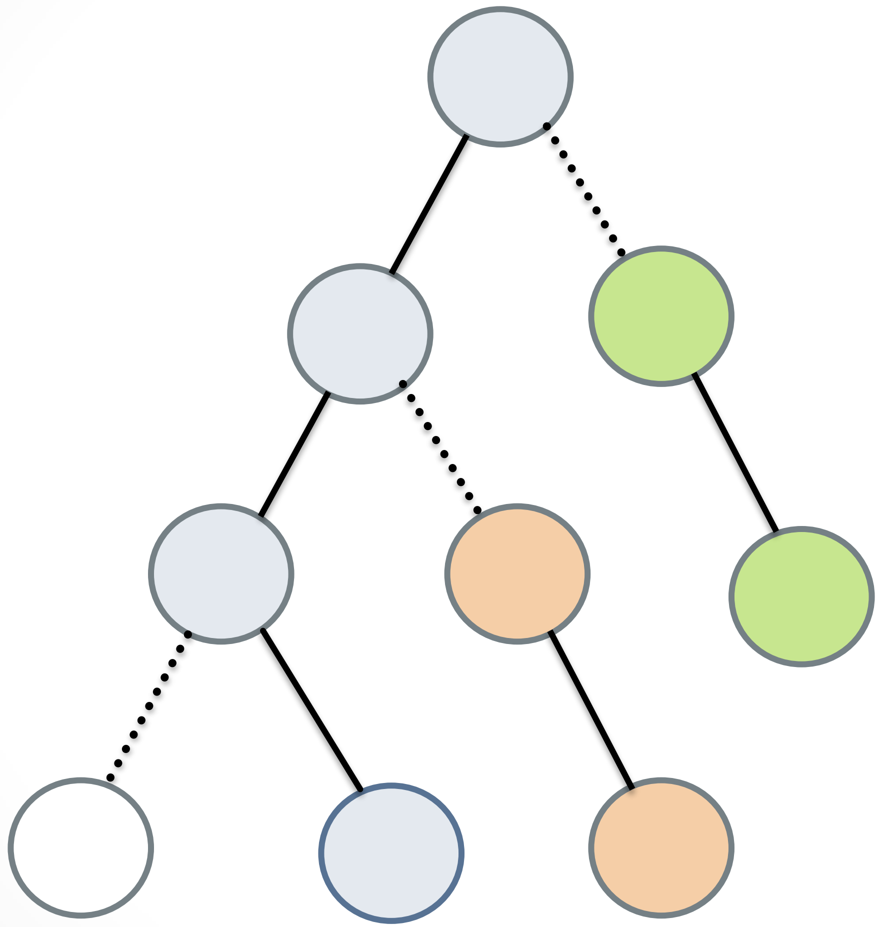
- 他の部分を無視して、今考えてる列にだけ注目する
- 先頭 k 文字を x にする、という変更が起こる
- 変更クエリは列を取り出すと扱いやすい？？

Heavy Light Decomposition

HL 分解

- 根付き木を列に分解するアルゴリズム
- 各頂点 v から根までのパスは高々 $\log N$ 個の列にしか属さない

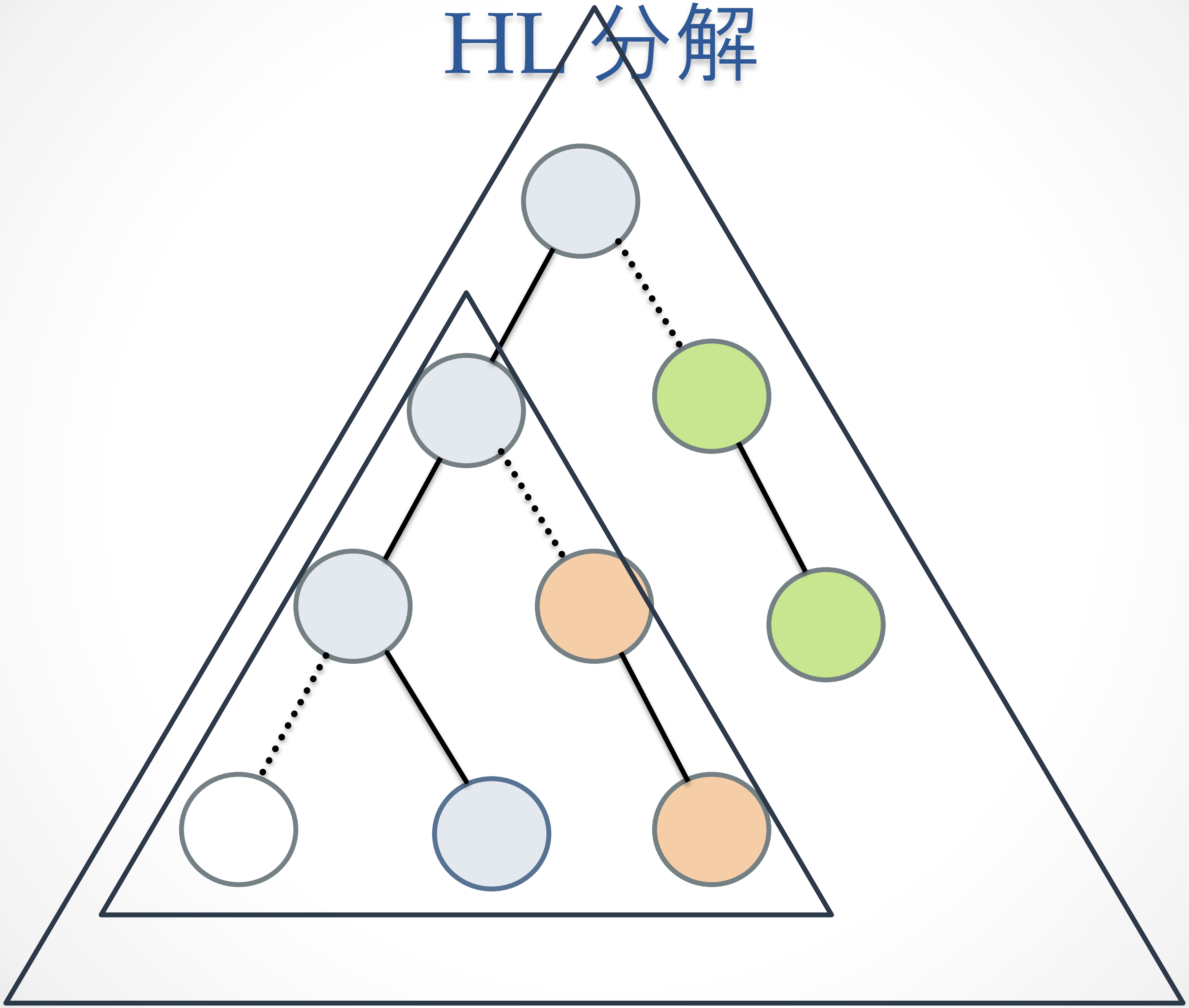
HL 分解



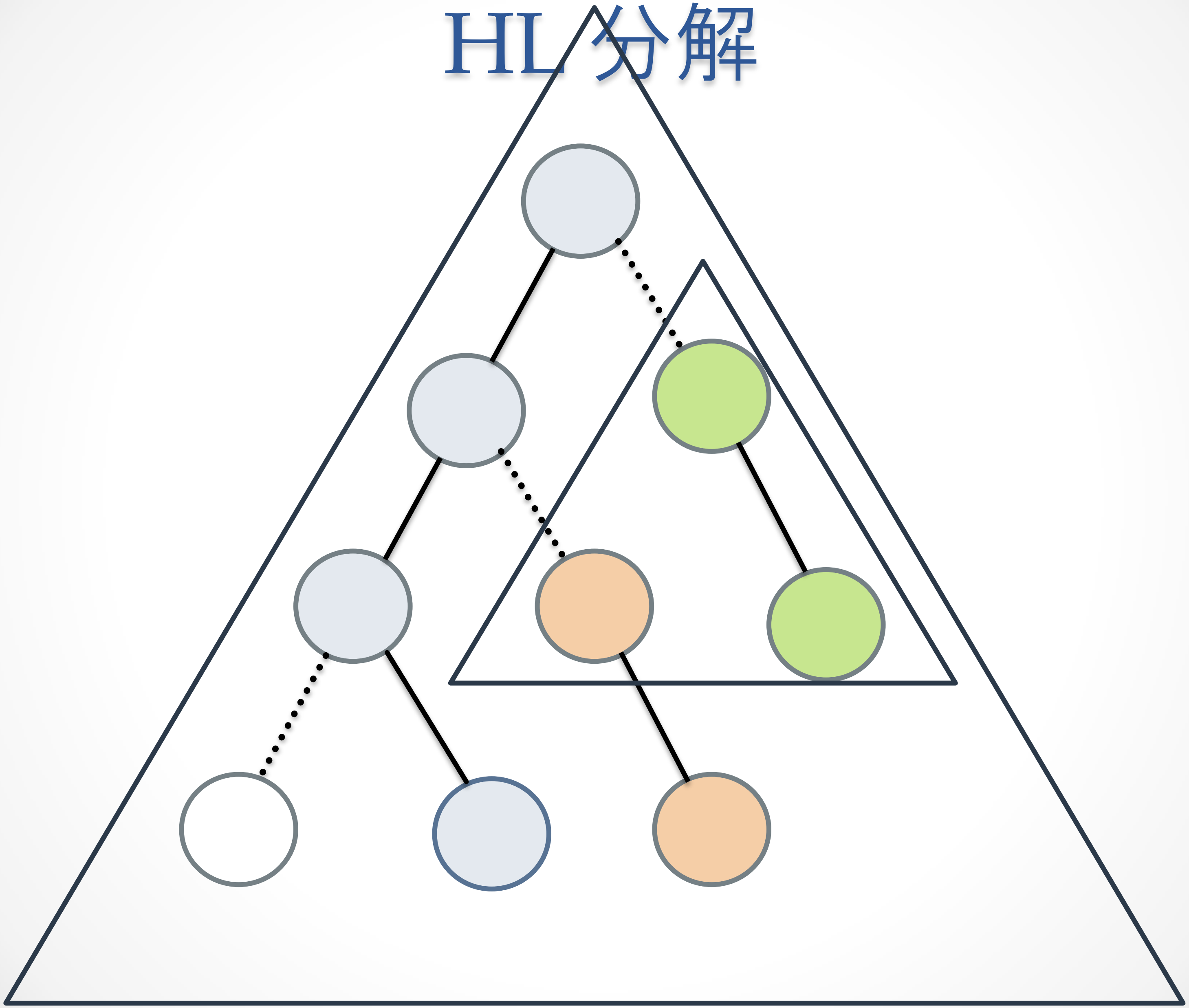
HL 分解 – 作り方

- 各頂点 v に対して、子供への辺を高々一つ決めればよい
- v の部分木サイズを S_v として、 $S_v \leqq 2 * S_u$ なる子供 u は高々一つ
- そのような u への辺を採用する

HL分解



HL分解



HL 分解 – 解析

- 「各頂点 v から根までのパスは高々 $\log N$ 個の列にしか属さない」という条件は成り立ってる？？

HL 分解 – 解析

- 「各頂点 v から根までのパスは高々 $\log N$ 個の列にしか属さない」という条件は成り立ってる？？
- 言い換えると、各頂点 v から根までのパスに、列に含まれない辺が高々 $\log N$ 個しか存在しないということ

HL 分解 – 解析

- 「各頂点 v から根までのパスは高々 $\log N$ 個の列にしか属さない」という条件は成り立ってる？？
- 言い換えると、各頂点 v から根までのパスに、列に含まれない辺が高々 $\log N$ 個しか存在しないということ
- 先の式を見れば、列に含まれないパスを子供から親へ上ると、部分木サイズが 2 倍になるから、確かにそうなる

ところで

- クエリごとに頂点を追加するけど、動的に HL 分解を管理することなんてできるの？？？

ところで

- クエリごとに頂点を追加するけど、動的に HL 分解を管理することなんてできるの？？？
- 最終的にできる木は先読みできるから、そこで HL 分解を作ればよい

今までの振り返り

- 0 が根の根付き木とその HL 分解が求まっている
- そこで、以下の操作を行う
- ある頂点 v から根までのパスまでの数列を得て、その inversion を求める
- そして、パス上の値をすべて一定の値に書き換える

あとは簡単

- 列の時の考察から、各列の更新は愚直にできる（ただし、同じ値は圧縮する必要がある）
- 各列での更新ができるなら、各列の更新前も分かる
- 圧縮した値を持っていても、inversion は計算できる

具体的に

- HL 分解を行い、各列ごとに「x が k 個連続している」というように圧縮して、値を管理しておく
- 頂点 v について操作を行うとき、v から根までのパスに属する列をすべて調べる
- 列ごとに、愚直に更新する
- 圧縮した列で inversion を求める

計算量解析

- 頂点 v を追加したとき、見る列の個数は $\log N$ 個
- それぞれの列で新しい値は 1 個入るだけだから、値が消えるのは全部合わせても $N \log N$ 回だけ
- inversion の計算は $O($ (消える個数) $\log($消える個数$)\ )$ だから、すべて合わせると $O(N \log^2 N)$
- 間に合う！！

得点分布