

# こわくない全方位木DP

MiuraMiuMlu(@\_sono\_8\_)

Hokkaido University M1

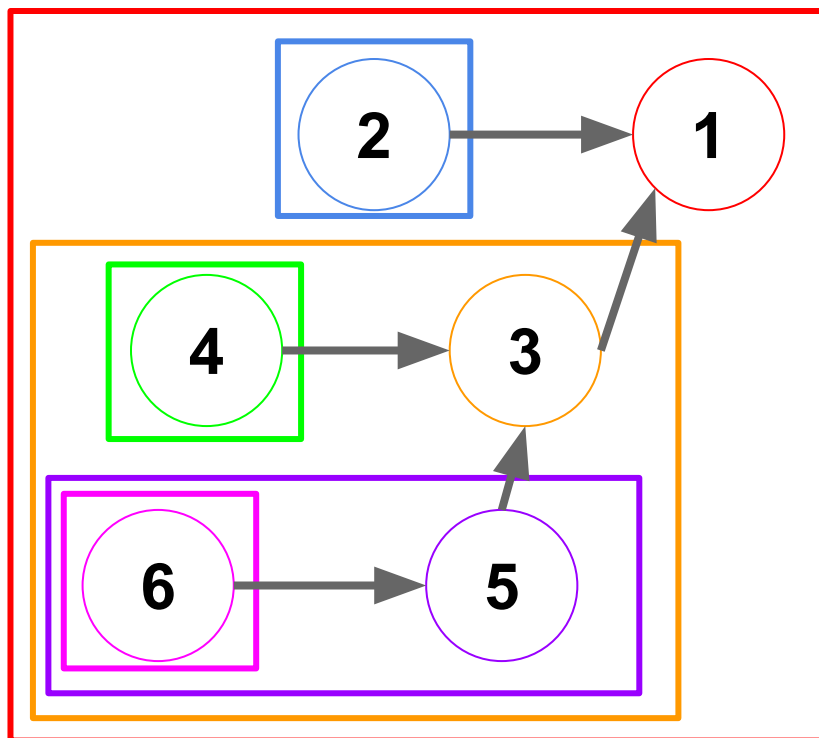
Nov 7, 2019

## 本日の目標

- 全方位木DPの性質を学ぶ
- 全方位木DPを実装できるようになる

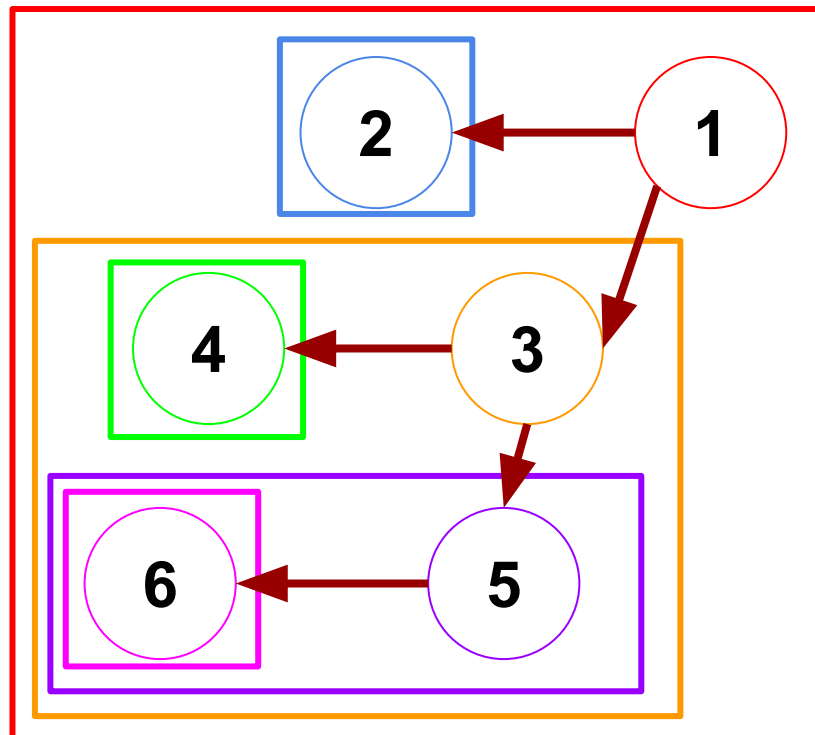
## [前提]木DPとは何か

- 各頂点を根とした**部分木の状態数**を保持
- **子から親に**状態数を伝搬させて更新 →  $O(N)$



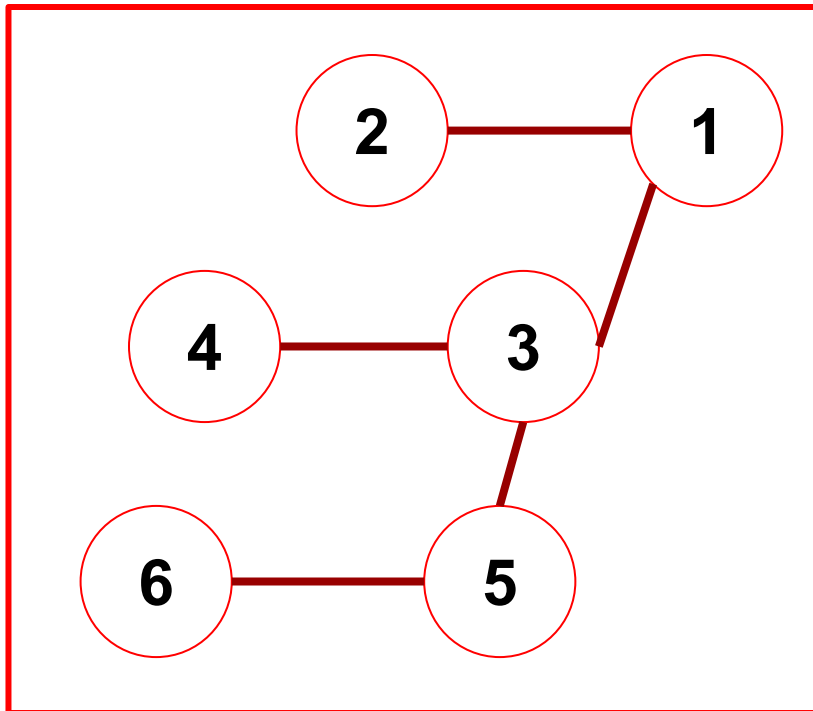
# 全方位木DPとは何か

- 各頂点を根とした**木全体の状態数**を保持
- 木DPの結果を利用
- **親から子**に状態数を伝搬させて更新 →  $O(N)$



# 何がうれしい？

- 各頂点を根とした状態数を $O(N)$ で計算できる



# 実践から学ぶ - 問題概要

## ・問題

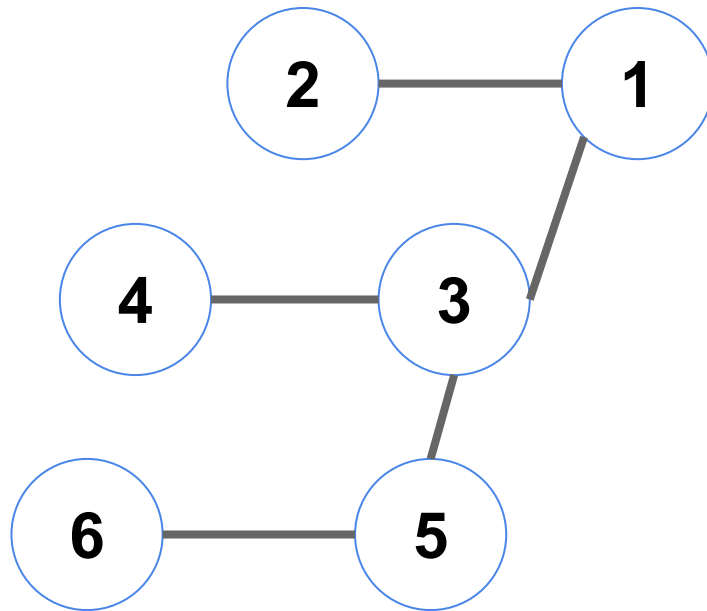
それぞれ1から $N$ までの番号が付いた $N$ 個の頂点からなる木が与えられる。各頂点について、その頂点から最も遠い頂点までの距離を求めよ。

## ・制約

$$1 \leq N \leq 100,000$$

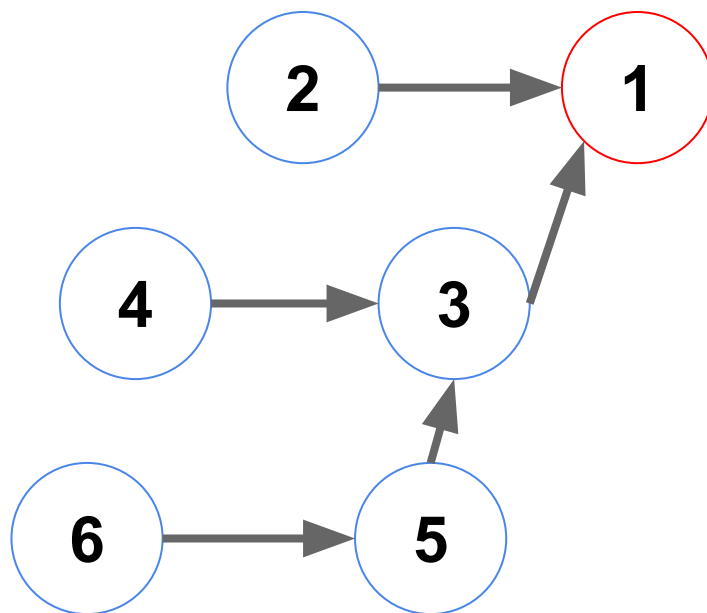
# 実践から学ぶ - イメージ

## ・例( $N=6$ )



# 実践から学ぶ - イメージ

## ・頂点1を根としたとき

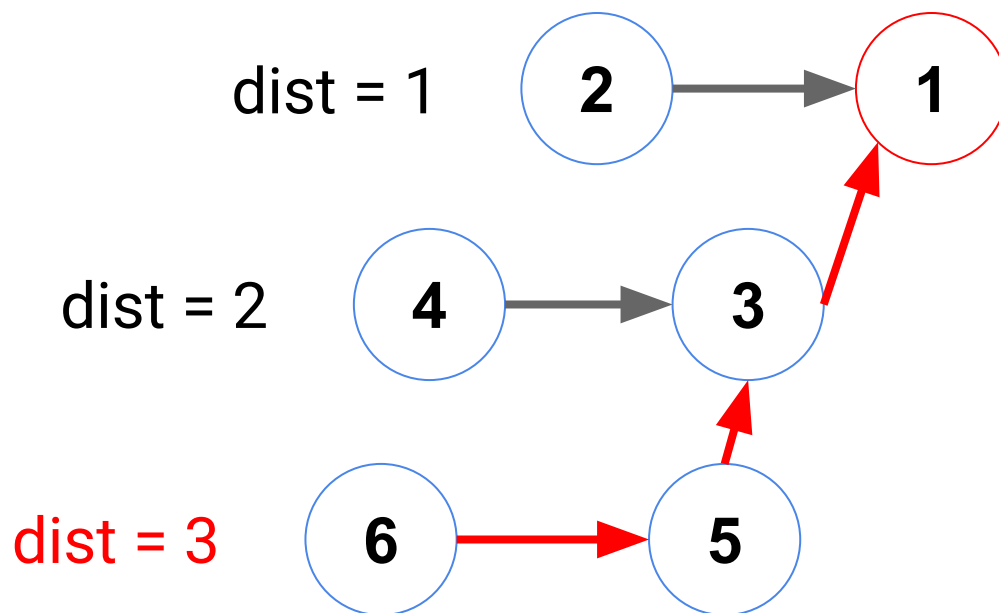


- 根からDFS



# 実践から学ぶ - イメージ

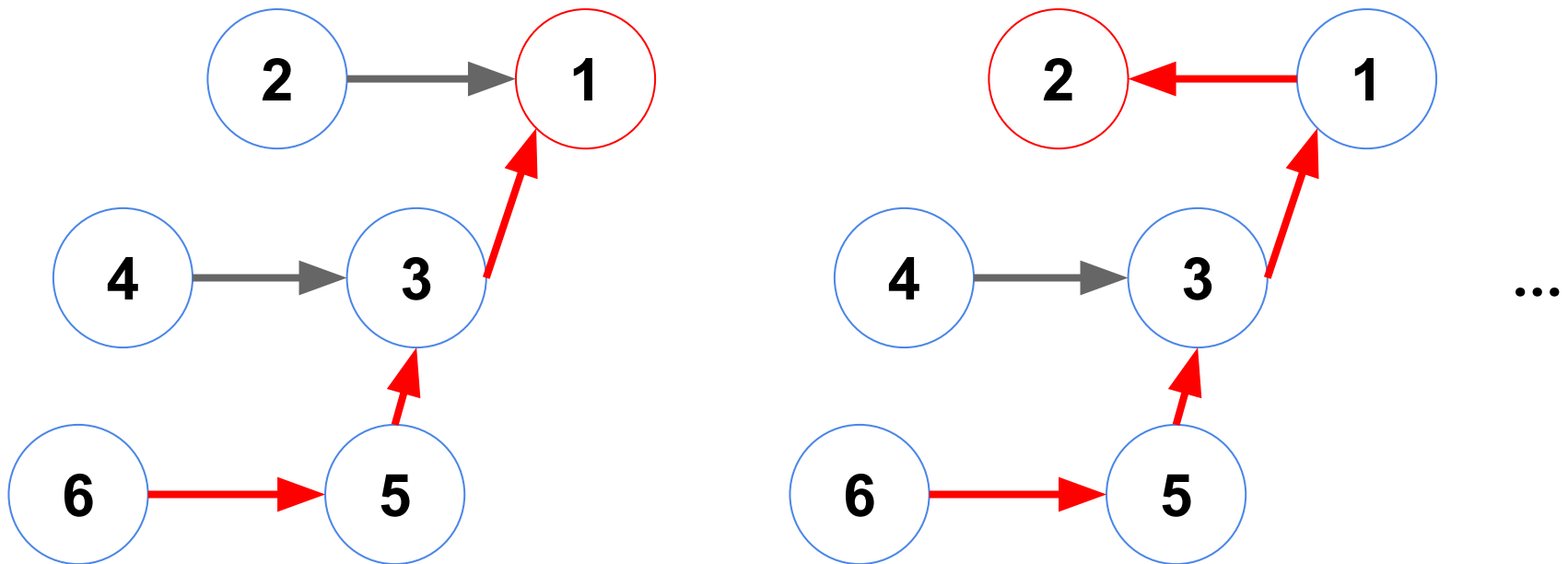
## ・頂点1を根としたとき



- 根からDFS
- $O(N)$

# 実践から学ぶ - イメージ

## ・すべての頂点を根としたとき



- 当然  $O(N^2)$   
→ 全方位木DPなら  $O(N)$  で解ける！

# 実践から学ぶ - 全方位木DPの考察方針

## ・全方位木DPで使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数の求め方を考察
3. 全方位木DPへの拡張

# 実践から学ぶ - 全方位木DPの考察方針

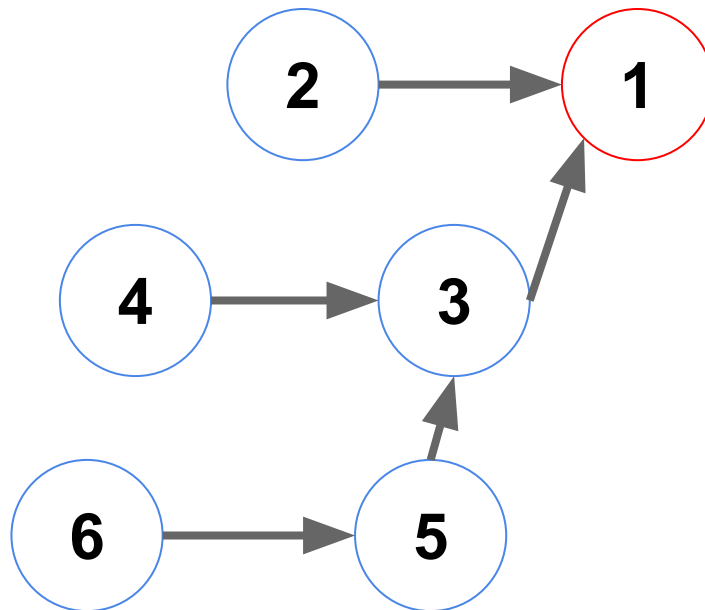
## ・全方位木DPで使えるような方針

1. **とりあえず木DP**
2. 親から伝搬させる状態数の求め方を考察
3. 全方位木DPへの拡張

# 実践から学ぶ - とりあえず木DP

## ・頂点1を根とする

$DP[i] :=$  頂点*i*を根とした部分木における最も遠い葉までの距離  
 $DP[i] = \max(DP[j]) + 1$  {  $j :=$  頂点*i*の子集合 }



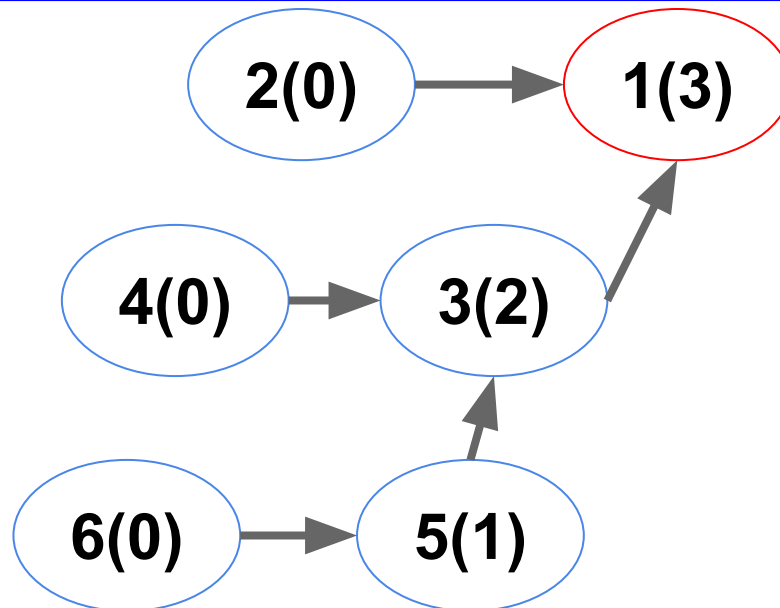
# 実践から学ぶ - とりあえず木DP

## ・頂点1を根とする

$DP[i] :=$  頂点*i*を根とした部分木における最も遠い葉までの距離

$DP[i] = \max(DP[j]) + 1$  {  $j :=$  頂点*i*の子集合 }

- メモ化再帰で求める



# 実践から学ぶ - 全方位木DPの考察方針

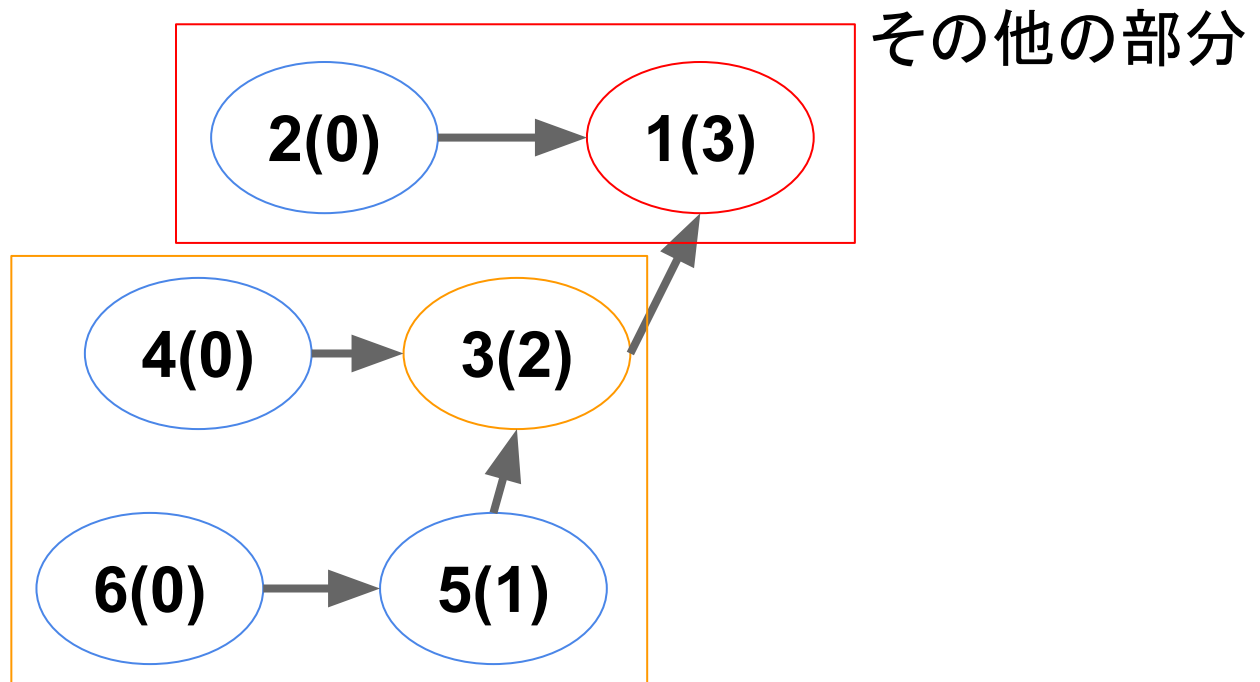
## ・全方位木DPで使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数の求め方を考察
3. 全方位木DPへの拡張

# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点3を根としたときを考える

頂点3の部分木ではなかつた部分を「その他の部分」とする.

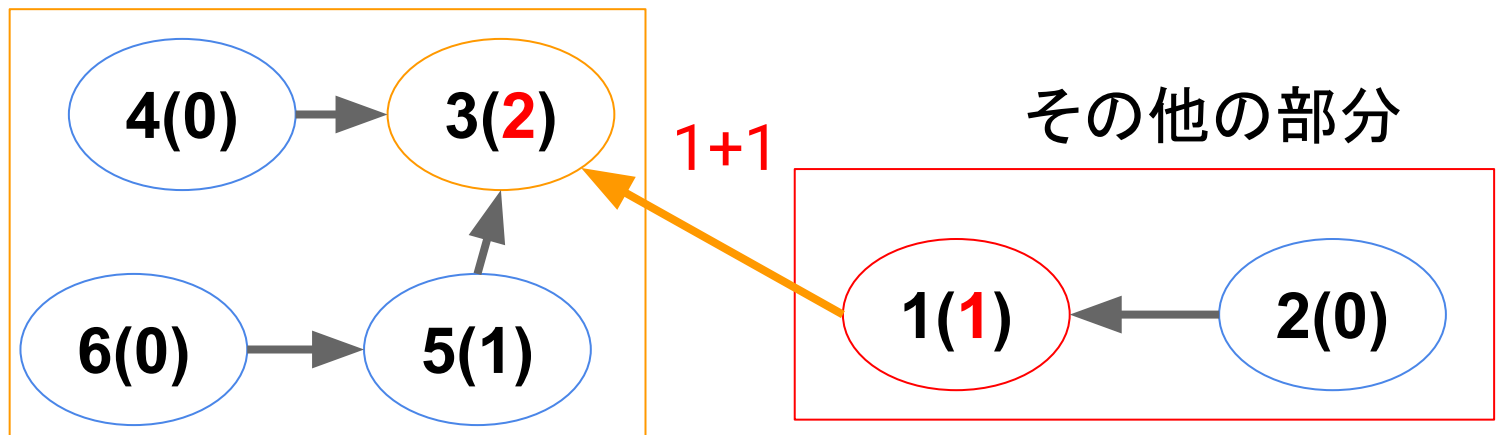




# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点3を根としたとき

「その他の部分」に関する木DPの状態数(最も遠い葉までの距離)を求め、頂点3における木DPの値を更新する。

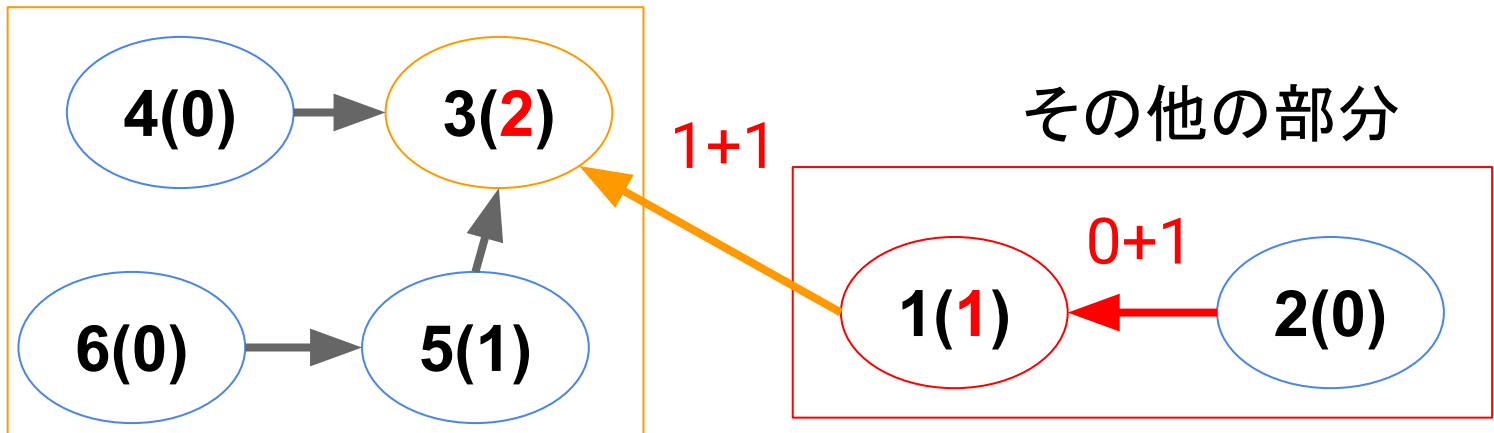


# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点3を根としたとき

「その他の部分」の最長距離を求めればよい.

→ 頂点1を根として「**その他の部分**」の最長距離を計算し, 頂点3に渡す

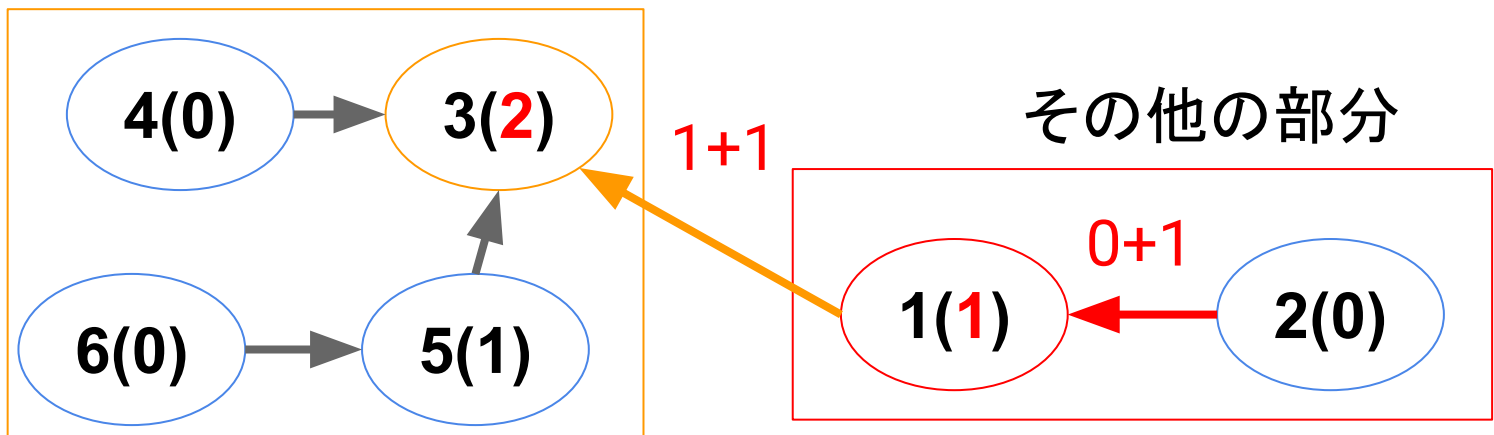


# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点3を根としたとき

「その他の部分」の最長距離を求めればよい。

→ 頂点1を根として**頂点3以外の子のDP結果**で最長距離を計算

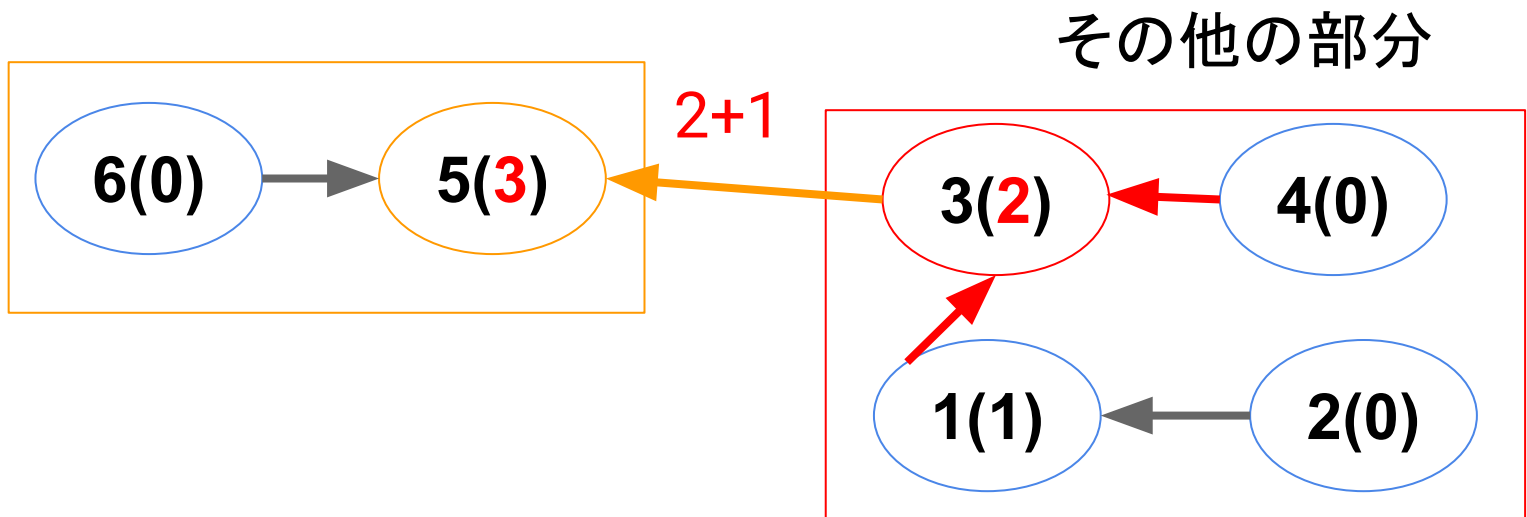


# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点5を根としたとき

同様に, 「その他の部分」の最長距離を求めればよい.

→ 頂点3を根として「**その他の部分**」の最長距離を計算し, 頂点5に渡す

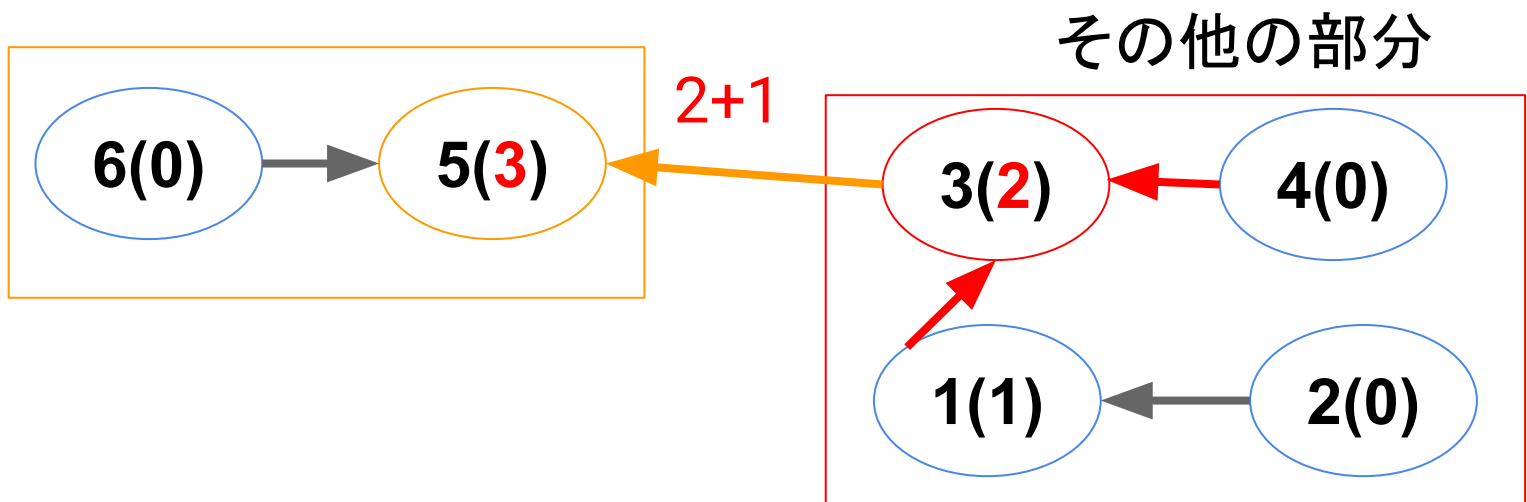


# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・頂点5を根としたとき

同様に, 「その他の部分」の最長距離を求めたい.

→ 頂点3を根として, **頂点5以外の子のDPの結果と親の計算結果を用いて**最長距離を計算



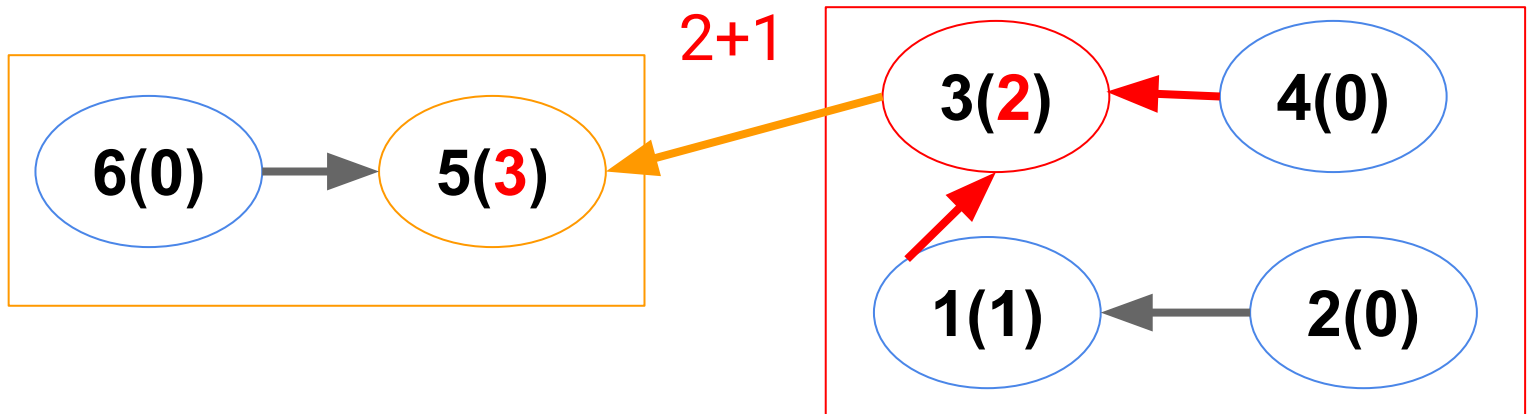
# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・伝搬における計算量

一回の伝搬につき, 渡す頂点を除く子の状態数と, 親から伝搬される状態数を用いる.

→ 最悪 $O(N)$ の計算量

## ・頂点3→頂点5への伝搬

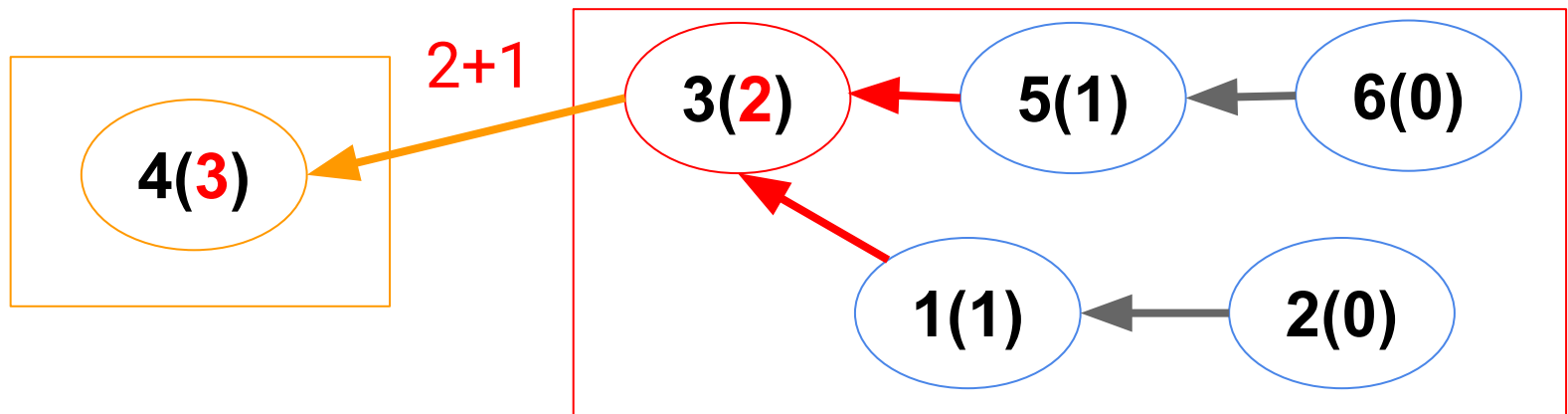


# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・伝搬における計算量

伝搬される値は子の個数だけ計算する必要がある  
→ 最悪 $O(N^2)$ かかってしまう.

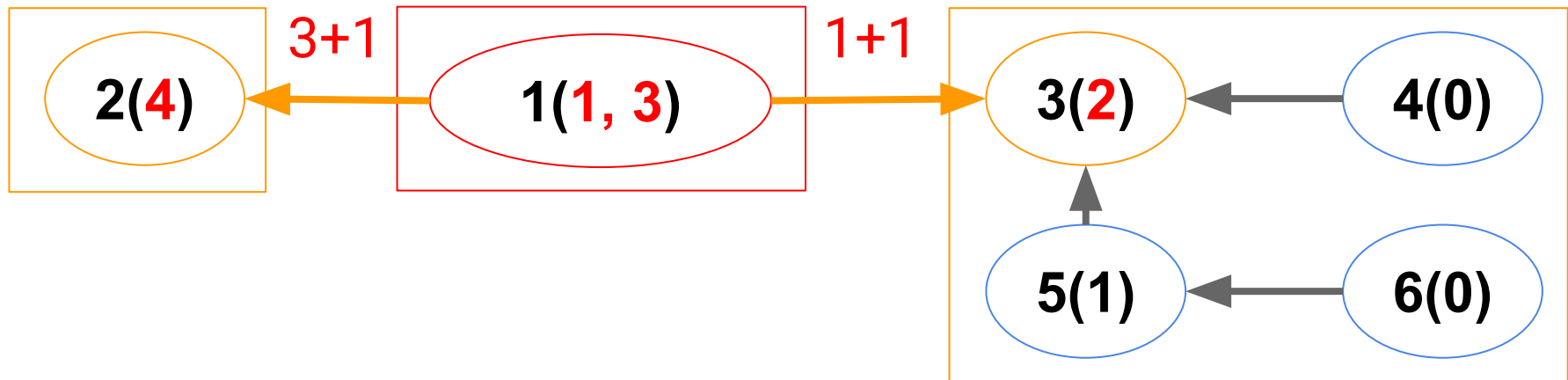
## ・頂点3→頂点4への伝搬



# 実践から学ぶ - 親から伝搬させる状態数を考える

## ・伝搬における計算量

考えてみると, 距離が「最大のもの」と「2番目に大きなもの」さえ把握できていれば「その他の部分」の最長距離はそのうちのどちらかなので $O(1)$ で求められる.





# 実践から学ぶ - 全方位木DPの考察方針

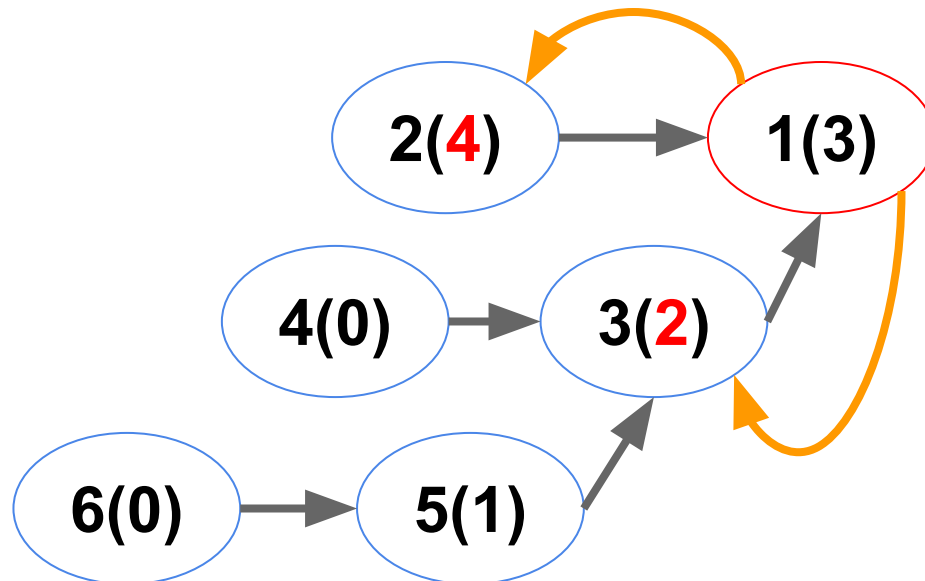
## ・全方位木DPで使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数の求め方を考察
3. 全方位木DPへの拡張

# 実践から学ぶ - 全方位木DP

## ・処理の流れ

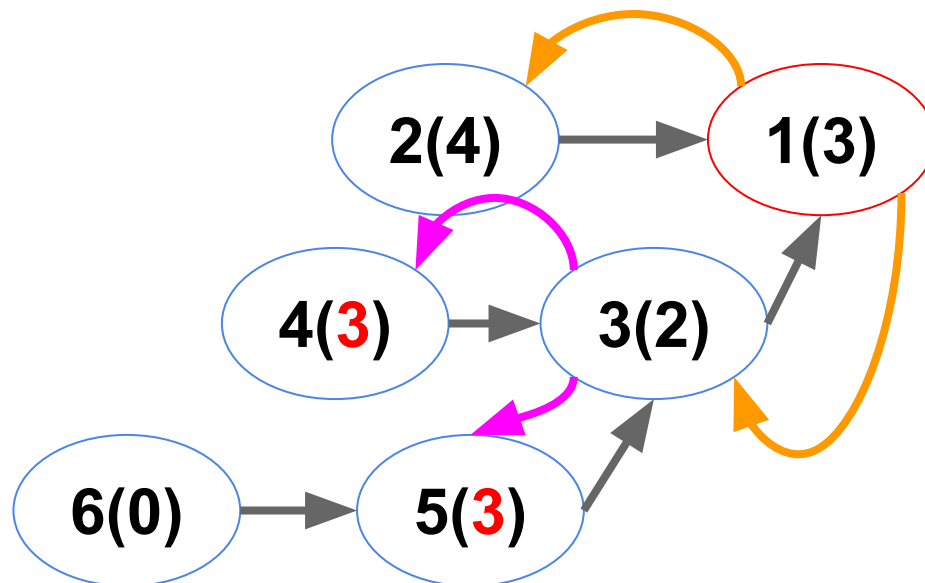
親からDFSで値を伝搬させていく。



# 実践から学ぶ - 全方位木DP

## ・処理の流れ

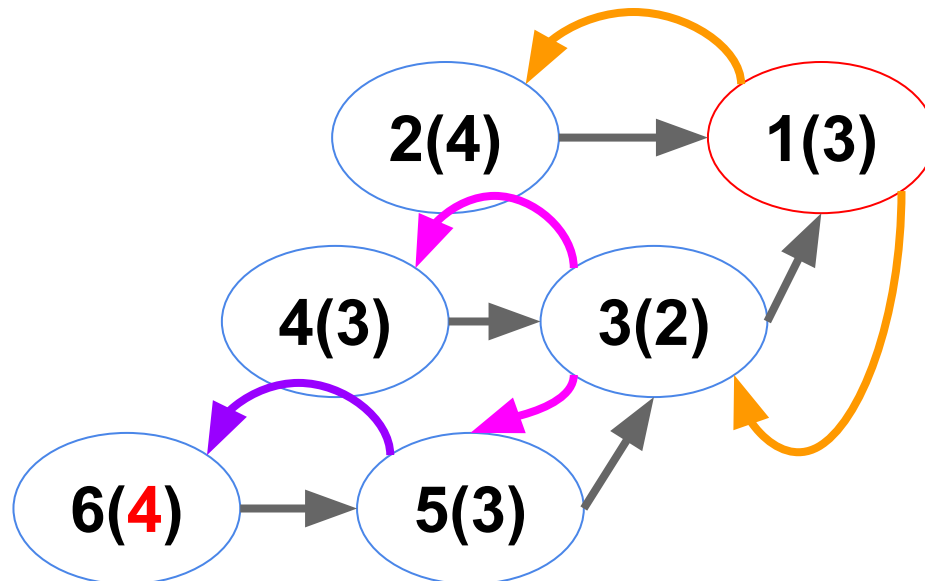
親からDFSで値を伝搬させていく。



# 実践から学ぶ - 全方位木DP

## ・処理の流れ

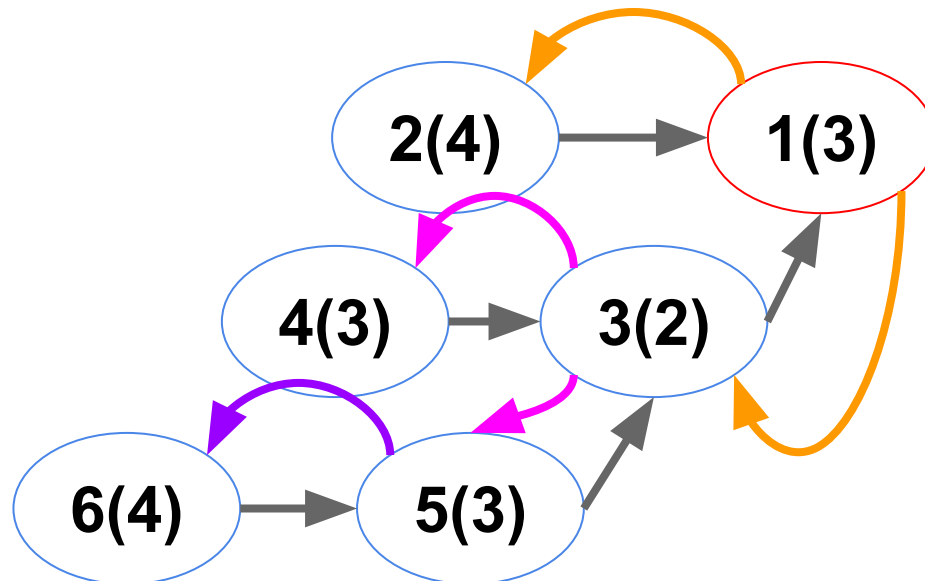
親からDFSで値を伝搬させていく。



# 実践から学ぶ - 全方位木DP

## ・処理の流れ

親からDFSで値を伝搬させていく。  
各ノードは渡す状態数の計算と伝搬で高々2回しか見ない。  
→ 全部で $O(N)$ の計算量。



## 実践から学ぶ - まとめ

1. とりあえず木DP  
→ 根を固定してメモ化再帰
2. 親から伝搬させる状態数の求め方を考察  
→ 「その他の部分」に関する状態数を求める
3. 全方位木DPへの拡張  
→ 木DPの根からDFSで状態数を伝搬

# 練習問題 Traffic Tree(AOJ)

## ・問題文

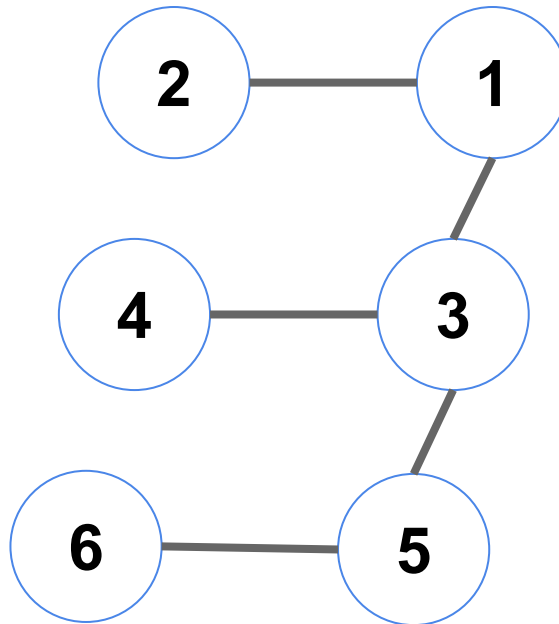
それぞれ1からNまでの番号が付いたN個の頂点が、N-1本の無向辺によって繋がれたグラフが与えられる。各頂点について、その頂点からスタートしてすべての頂点を訪れるための最短のステップ数を出力せよ。

## ・制約

$$1 \leq N \leq 100,000$$

# 練習問題 Traffic Tree(AOJ)

・例(サンプル2, N=6)

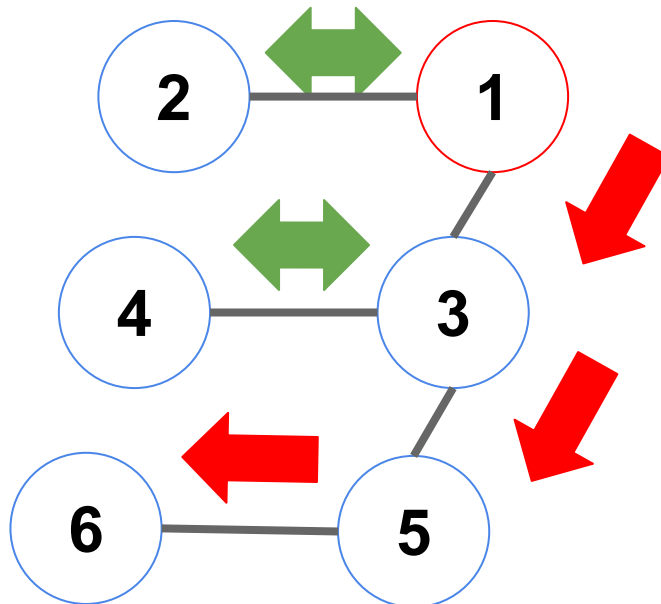




# 練習問題 Traffic Tree(AOJ)

## ・頂点1を根とした場合

全7ステップ。

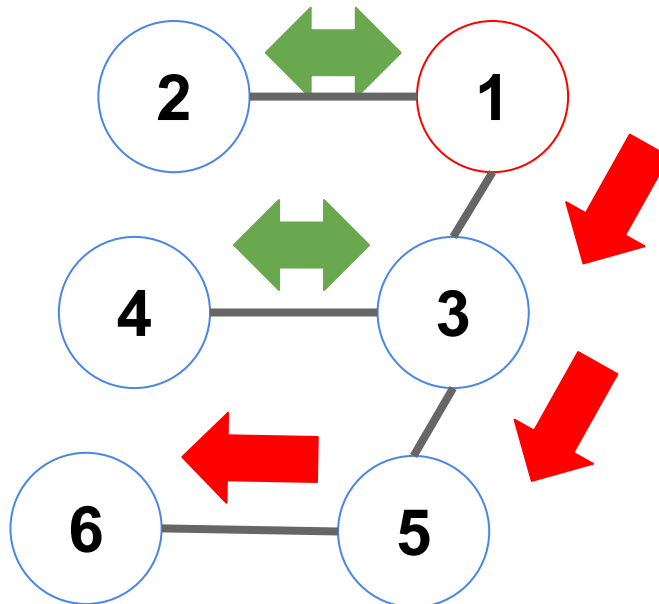


# 練習問題 Traffic Tree(AOJ)

## ・性質

葉までのパスについて, 往路のみのものが1つ, それ以外は往復する.

→ 往路のみのパスをできるだけ長くしたい.

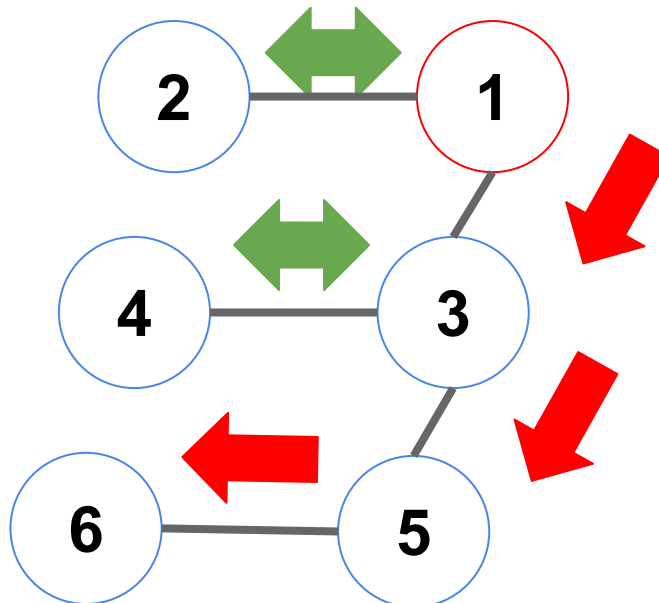


# 練習問題 Traffic Tree(AOJ)

## ・実装

往路のみのパスの最長距離は, 最遠の葉までの距離に等しい  
→ 求めるステップ数 $S$ は以下で求められる.

$$S = (\text{辺の総数}) \times 2 - (\text{最遠の葉までの距離})$$



# 練習問題 Traffic Tree(AOJ)

## ・実装

全方位木DPで各ノードを根とした場合の最も遠い葉までの距離を  
求める。  
すべてのノードについてステップ数を入力する。

## ・実装例

<http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3966957>  
(なおC#)

# 練習問題2 Driving on a Tree(AtCoder)

## ・問題文

N頂点の木が与えられ, 以下の操作を繰り返す.

- 隣り合った頂点に動く. ただし, 同じ頂点を2度通ってはいけない.
- 動ける頂点がない場合, そこで操作は終了となる.
- どこに動くかは等確率にランダムに選ぶ.

全ての頂点について, その頂点を始点とした場合の操作回数の期待値を求めよ.

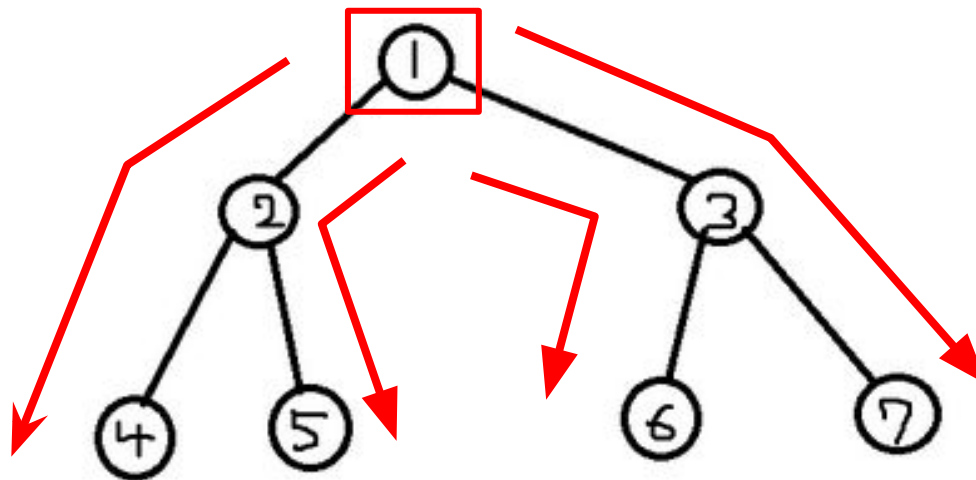
## ・制約

$$1 \leq N \leq 150,000$$

## 練習問題2 Driving on a Tree(AtCoder)

### ・入力例4 (N=7)

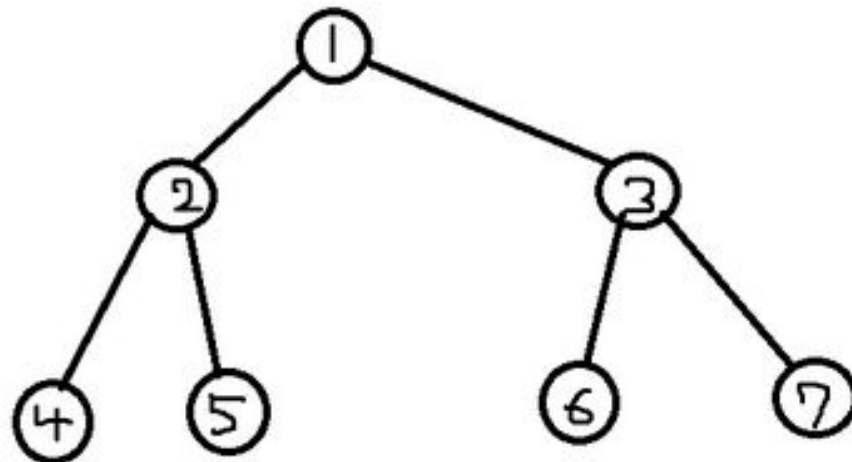
頂点1を始点とすると, 以下のような操作が考えられる.  
いずれも $1/4$ の確率で行われ, 操作回数は2回.  
→ 操作回数の期待値は  $1/4 \times 2 \times 4 = 2$ .



## 練習問題2 Driving on a Tree(AtCoder)

### ・考察

- 期待値なのでDPで求められそう.
- 全ての頂点について出力しなければならない.  
→ 全方位木DPが使える!



## 練習問題2 Driving on a Tree(AtCoder)

### ・全方位木DP全般で使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数を考察
3. 全方位木DPへの拡張



## 練習問題2 Driving on a Tree(AtCoder)

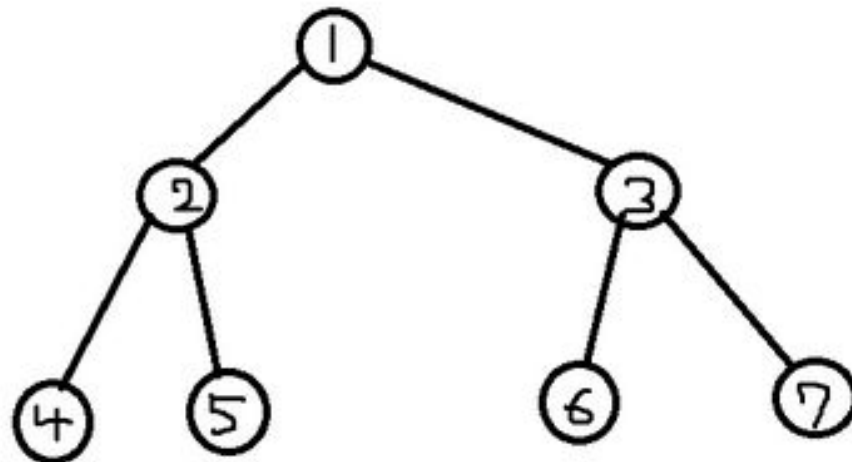
### ・全方位木DP全般で使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数を考察
3. 全方位木DPへの拡張

## 練習問題2 Driving on a Tree(AtCoder)

### ・頂点1を根として木DP

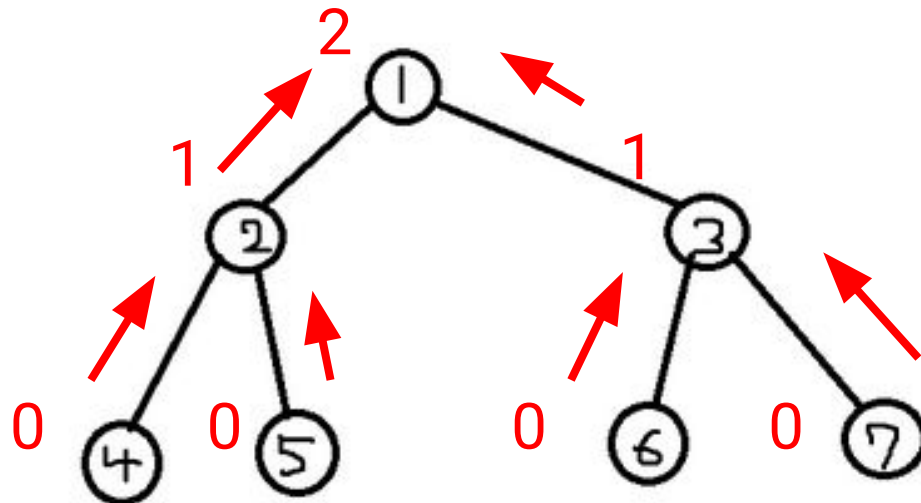
$DP[i] :=$  頂点 $i$ を根とした部分木における操作回数の期待値  
 $DP[i] = \text{avg}(DP[j] + 1) \{j := \text{頂点}i\text{の子集合}\}$



## 練習問題2 Driving on a Tree(AtCoder)

### ・頂点1を根として木DP

$DP[i] :=$  頂点 $i$ を根とした部分木における操作回数の期待値  
 $DP[i] = \text{avg}(DP[j] + 1) \{j := \text{頂点}i\text{の子集合}\}$



## 練習問題2 Driving on a Tree(AtCoder)

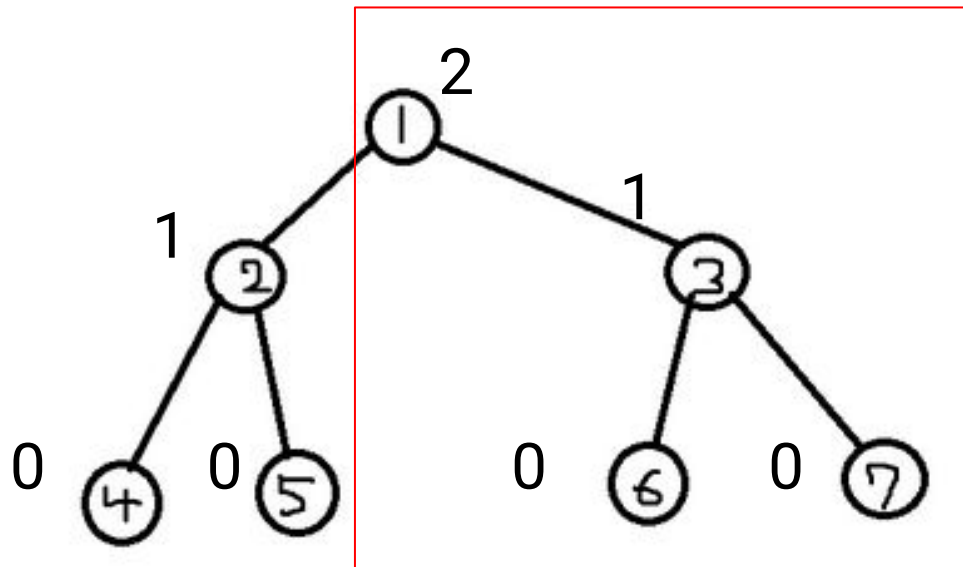
### ・全方位木DP全般で使えるような方針

1. とりあえず木DP
2. 親から伝搬させる状態数を考察
3. 全方位木DPへの拡張

## 練習問題2 Driving on a Tree(AtCoder)

### ・頂点2を根とした場合の伝搬される状態数

親から伝搬する状態数は、「その他の部分」の期待値.

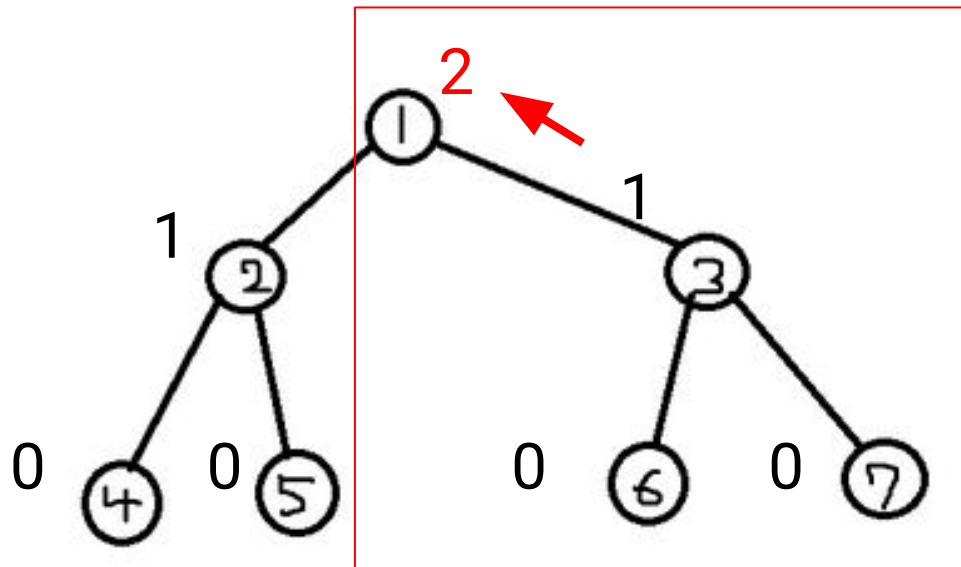


## 練習問題2 Driving on a Tree(AtCoder)

### ・頂点2を根とした場合の伝搬される状態数

「その他の部分」の期待値は、親である頂点1の期待値から頂点2を根とする部分木の情報を除いたもの。

これは親の期待値と木DPの結果から $O(1)$ で計算可能。

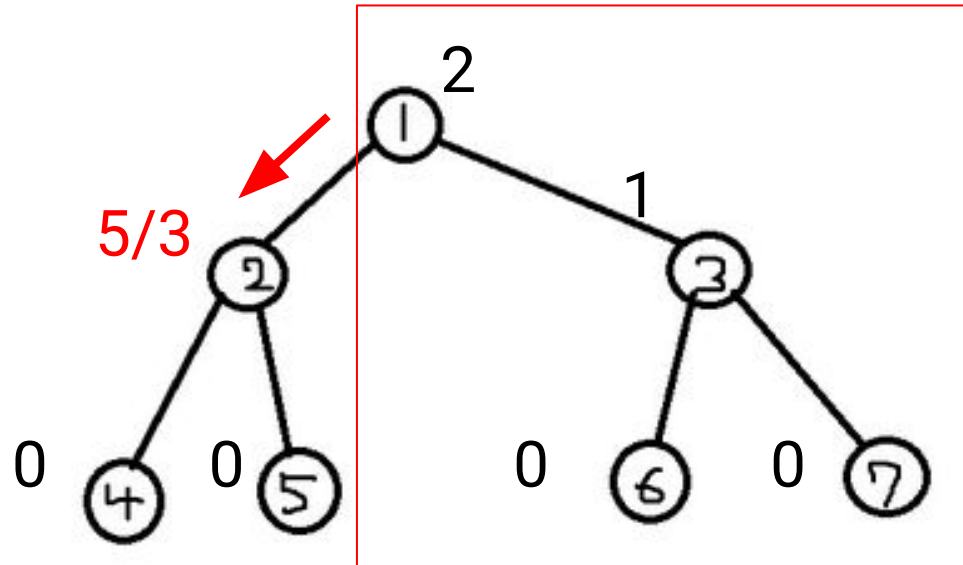


その他の部分

## 練習問題2 Driving on a Tree(AtCoder)

### ・頂点2を根とした場合の伝搬される状態数

「その他の部分」の期待値で頂点2の期待値を更新.



## 練習問題2 Driving on a Tree(AtCoder)

### ・全方位木DP全般で使えるような方針

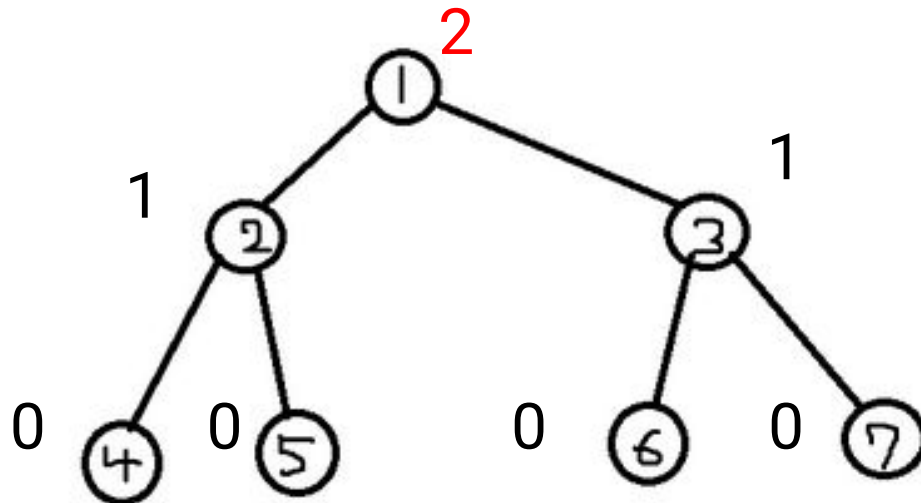
1. とりあえず木DP
2. 親から伝搬させる状態数を考察
3. 全方位木DPへの拡張



# 練習問題2 Driving on a Tree(AtCoder)

## ・全方位木DP

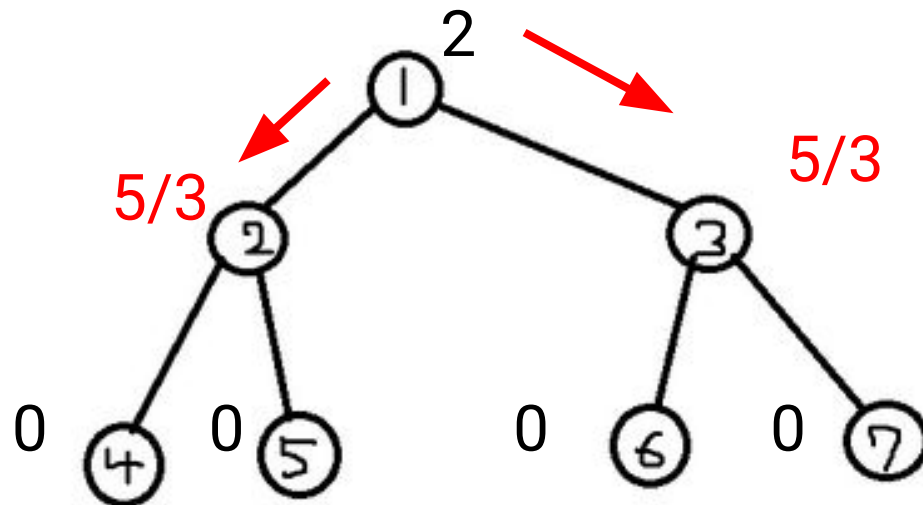
頂点1からDFS



# 練習問題2 Driving on a Tree(AtCoder)

## ・全方位木DP

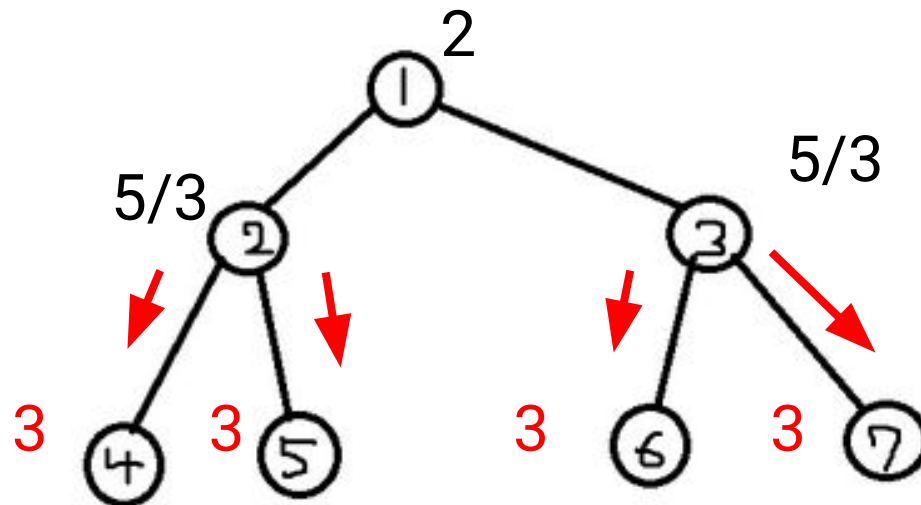
頂点1からDFS



# 練習問題2 Driving on a Tree(AtCoder)

## ・全方位木DP

頂点1からDFS



## 練習問題2 Driving on a Tree(AtCoder)

### ・実装

全方位木DPで各頂点に関する期待値を計算する.

### ・実装例

<https://atcoder.jp/contests/s8pc-4/submissions/8290101>

(なおC#)

# まとめ

## ・全方位木DPの流れ

1. とりあえず木DP  
→ 根を固定してメモ化再帰
2. 親から伝搬させる状態数の求め方を考察  
→ 「その他の部分」に関する状態数を求める
3. 全方位木DPへの拡張  
→ 木DPの根からDFSで状態数を伝搬

# 参考

## ・解説

- [+全方位木DP+について - ei1333の日記](#)
- [全方位木DP \(るま式\) | るまライブラリ](#)

## ・扱った問題

- [Traffic Tree](#)
- [D - Driving on a Tree](#)