

飛行機旅行 解説

JOI 2021/2022 春季トレーニング合宿

米田 優峻 / E869120

- 競技プログラミングでは、通常 **Batch 型課題**（出力の正誤によって採点される）が出題されます。
- しかしながら、情報オリンピックでは**通信型課題**などの特殊な形式が出題されることがあります。

通信型課題って
一体なに？

- 2つのプログラムを書く
- それらを問題で指定された形式でやりとりさせることによって、正解を導き出す。



Program 1

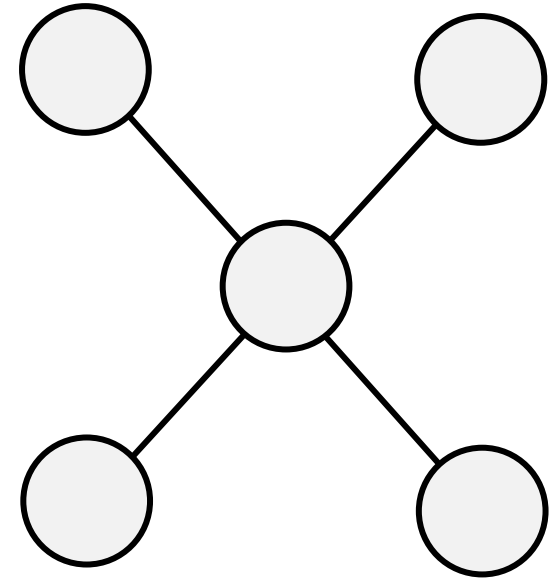


Program 2

Communication Task

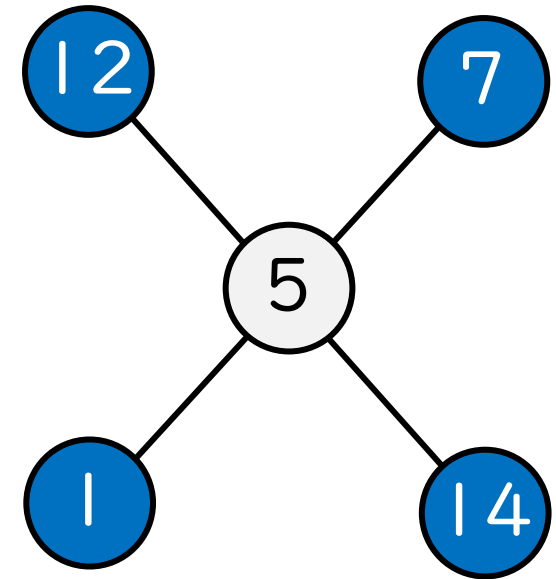
とも呼ばれる

0 N 頂点の木で表される航空路線図がある (Ali だけが知る)



0 N 頂点の木で表される航空路線図がある (Ali だけが知る)

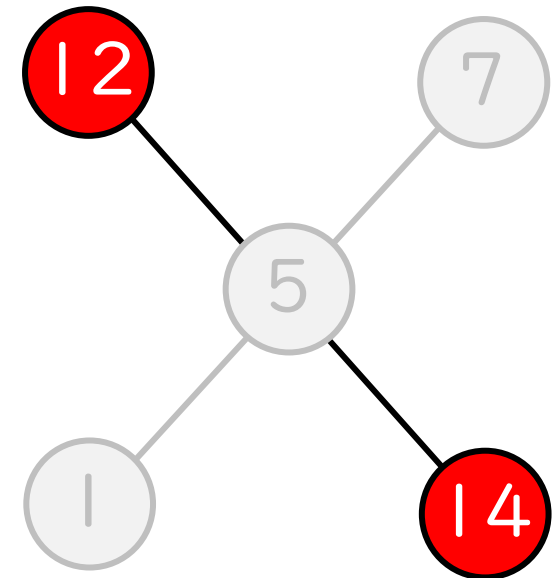
1 まず Ali は各頂点に ID を付ける



0 N 頂点の木で表される航空路線図がある (Ali だけが知る)

1 まず Ali は各頂点に ID を付ける

2 Benjamin は頂点 u, v の ID を得る



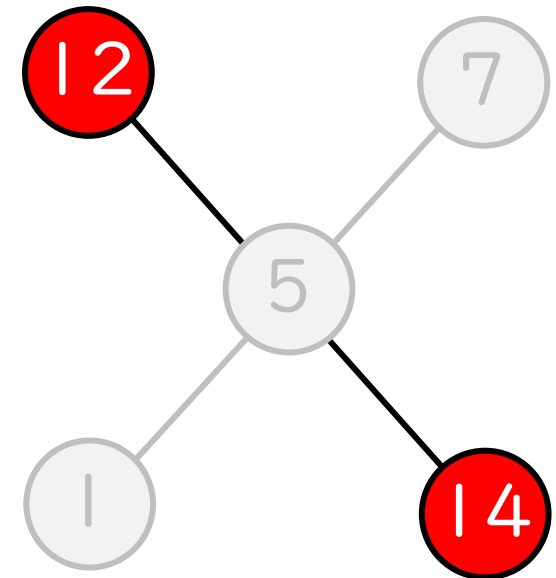
0 N 頂点の木で表される航空路線図がある (Ali だけが知る)

1 まず Ali は各頂点に ID を付ける

2 Benjamin は頂点 u, v の ID を得る

3 2 人が通信をして

Benjamin が 2 頂点間の距離※を答える



※距離 = 最短経路の長さ。

満点を取るためには…？

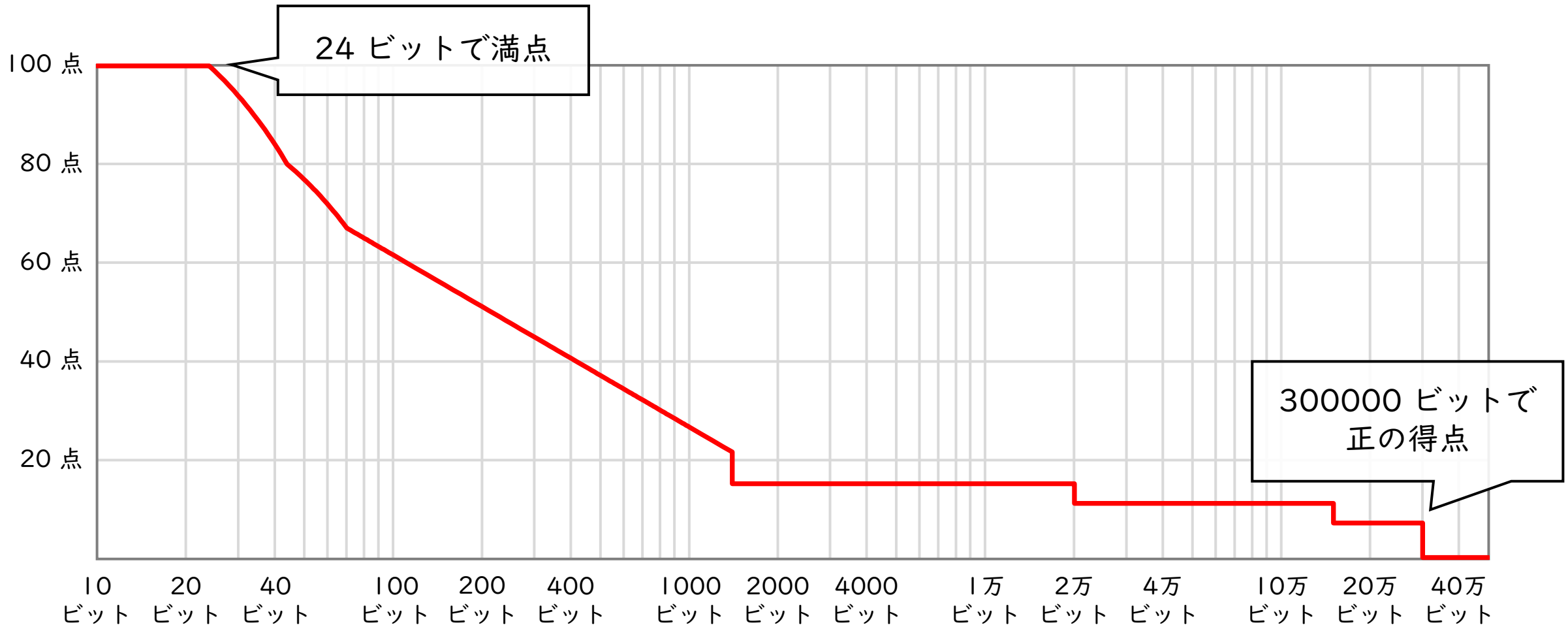
Benjamin→Ali

20 ビット

Ali→Benjamin

24 ビット以内

※以降は解説の都合上、「文字数」を「ビット数」として表します



Ali → Benjamin の最大ビット数 (文字数)

Stage 1

20000 ビットまでの解法

Ali が Benjamin に
木の情報を全部送ることを考える

必要なビット数は？

送るべき個数

20000個

辺の数は約 10000 本
それぞれの辺について
端点の番号を送るので
 $10000 \times 2 = 20000$ 個

必要なビット数は？

送るべき個数

1 個当たり

20000 個 × 14 ビット

辺の数は約 10000 本
それぞれの辺について
端点の番号を送るので
 $10000 \times 2 = 20000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14} - 1 = 16383$ までを
表せるので十分

必要なビット数は？

送るべき個数

1 個当たり

$$20000 \text{ 個} \times 14 \text{ ビット} = 280000 \text{ ビット}$$

辺の数は約 10000 本
それぞれの辺について
端点の番号を送るので
 $10000 \times 2 = 20000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14} - 1 = 16383$ までを
表せるので十分



必要なビット数は？

送るべき個数

1 個当たり

7

ここまでの点数：**7**点

辺の数は約 10000 本
それぞれの辺について
端点の番号を送るので
 $10000 \times 2 = 20000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14} - 1 = 16383$ までを
表せるので十分

Q. Benjamin は頂点番号 X, Y を
両方送れるのか？



頂点番号 X, Y を両方送るには？

頂点番号 X は 10000 まで ▶ 14 ビット

頂点番号 Y は 10000 まで ▶ 14 ビット



頂点番号 X, Y を両方送るには？

頂点番号 X は 10000 まで ▶ 14 ビット

頂点番号 Y は 10000 まで ▶ 14 ビット



合計 28 ビット

20 ビット制限に収まらない



点解法

頂点番号 X, Y を両方送るには？

頂点番号 X は 10000 不可能だが 4 ビット

頂点番号 Y は 10000 どちらか片方なら送れる



合計 28 ビット

20 ビット制限に収まらない

Benjamin が頂点番号 X を送り

Ali が番号 X から各頂点への距離を送る



点解法

Ali は何ビット必要か？

送るべき個数

10000個

各頂点までの距離を
全部送るので
頂点数 $N=10000$ 個

Ali は何ビット必要か？

送るべき個数

1 個当たり

10000 個 × 14 ビット

各頂点までの距離を
全部送るので
頂点数 $N=10000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14}-1=16383$ までを
表せるので十分



Ali は何ビット必要か？

送るべき個数

1 個当たり

$$10000 \text{ 個} \times 14 \text{ ビット} = 140000 \text{ ビット}$$

各頂点までの距離を
全部送るので
頂点数 $N=10000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14}-1=16383$ までを
表せるので十分



点解法

Aliは何ビット必要か？

送るべき個数

1個当たり

ここまでの点数 := 140000点

各頂点までの距離を
全部送るので
頂点数 $N=10000$ 個

頂点番号は 10000 まで
14 ビットあれば
 $2^{14}-1=16383$ までを
表せるので十分

7 点解法では木の情報を 280000 ビットかけて送ったが...

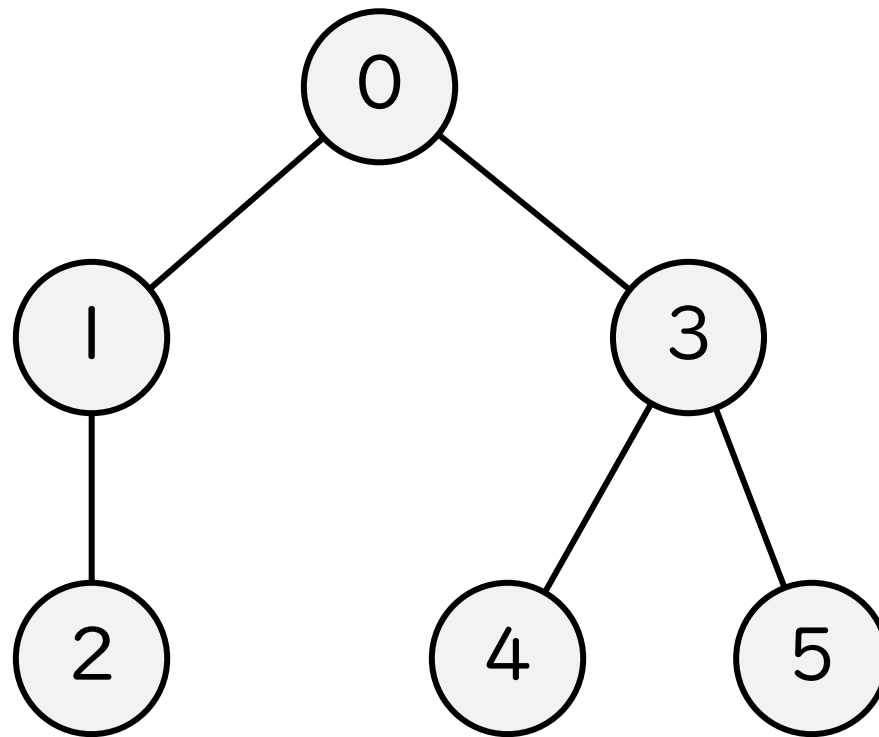
実は 2 万ビットで送れる

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

Ali が送る方法

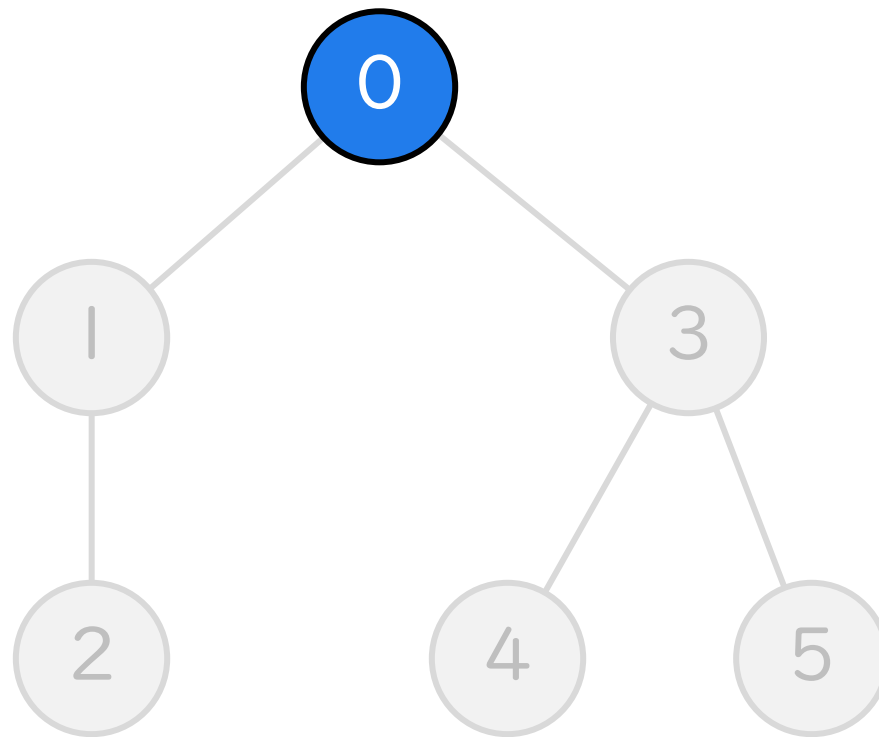
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：

Ali が送る方法

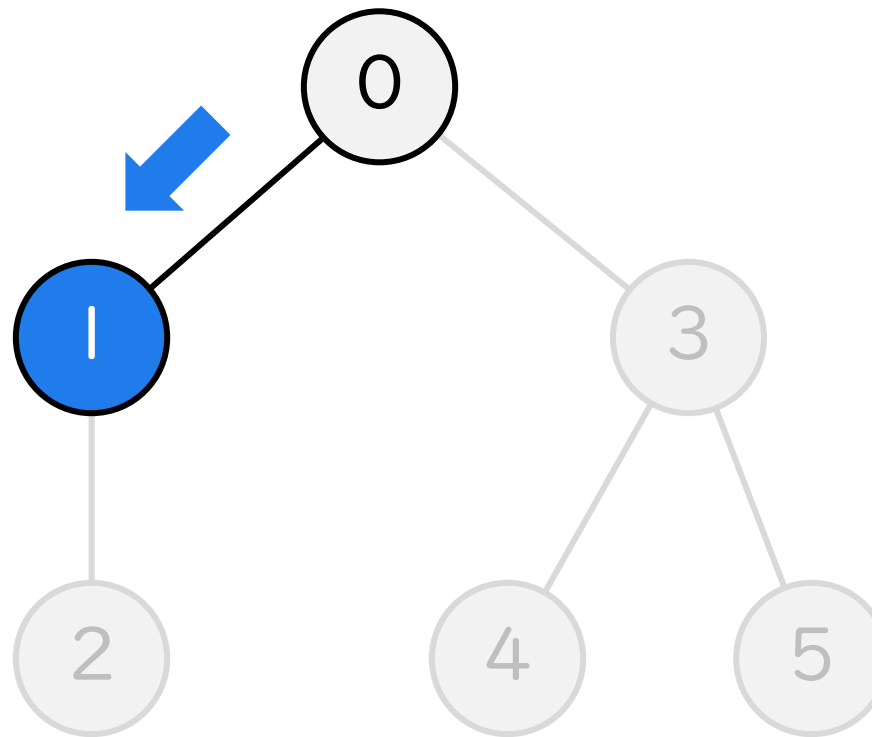
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：

Ali が送る方法

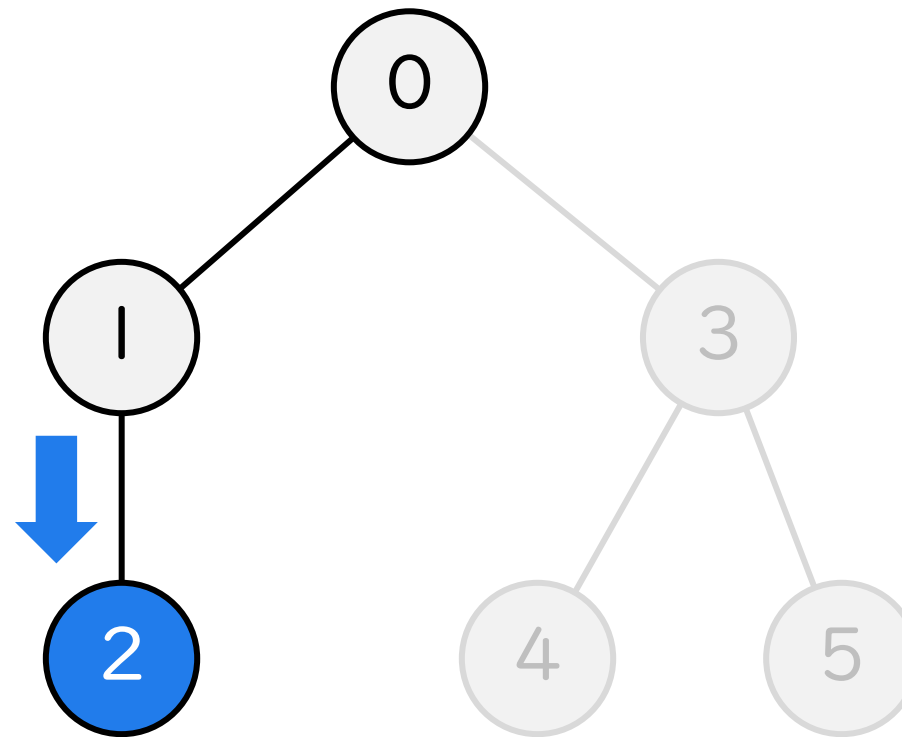
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列： 0

Ali が送る方法

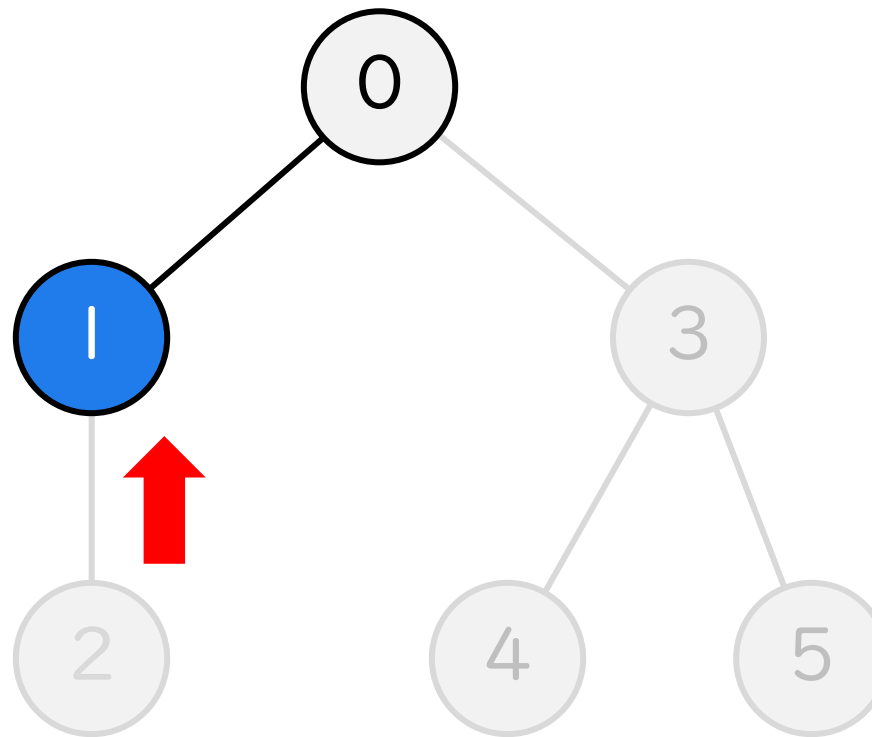
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列： 00

Ali が送る方法

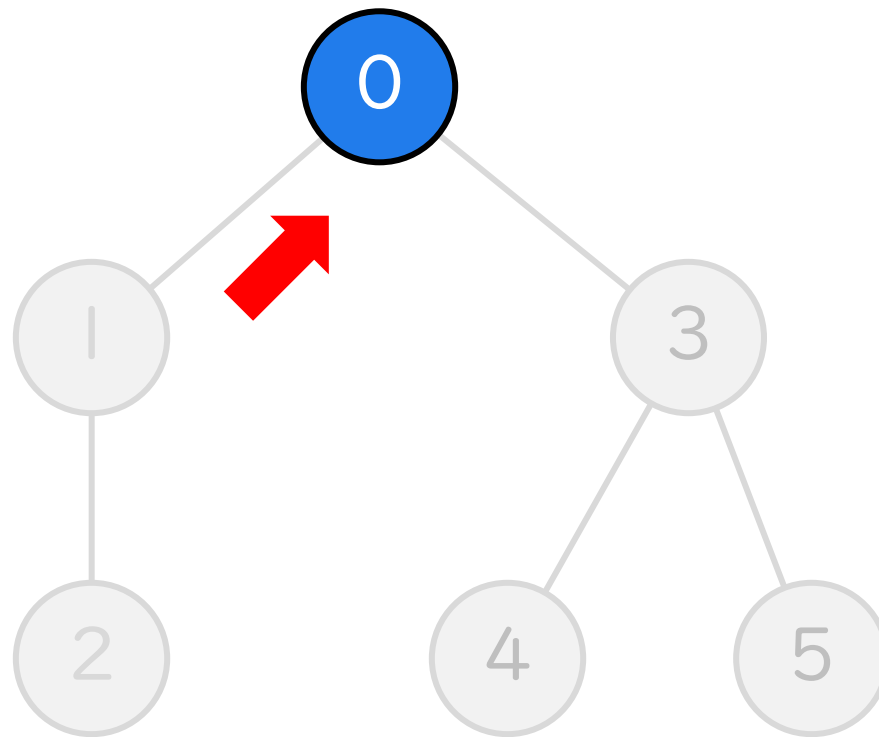
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：001

Ali が送る方法

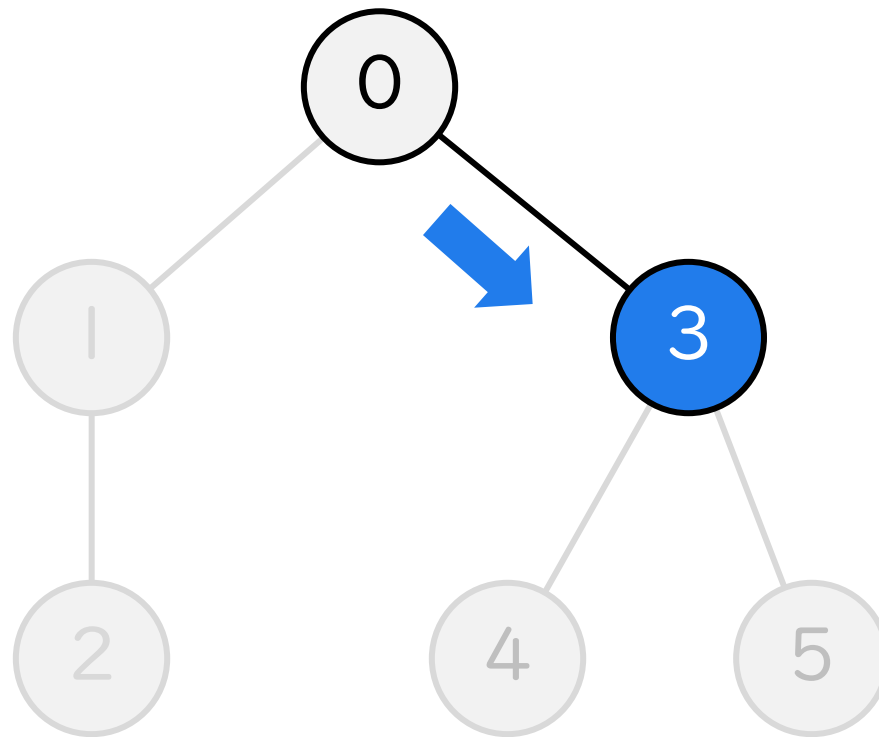
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：0011

Ali が送る方法

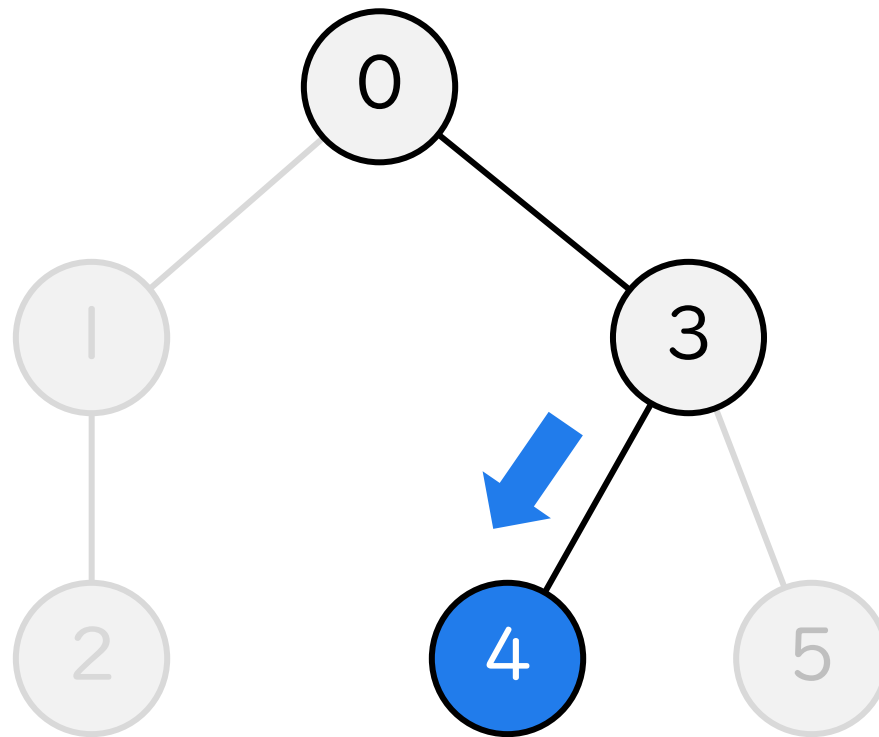
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：00110

Ali が送る方法

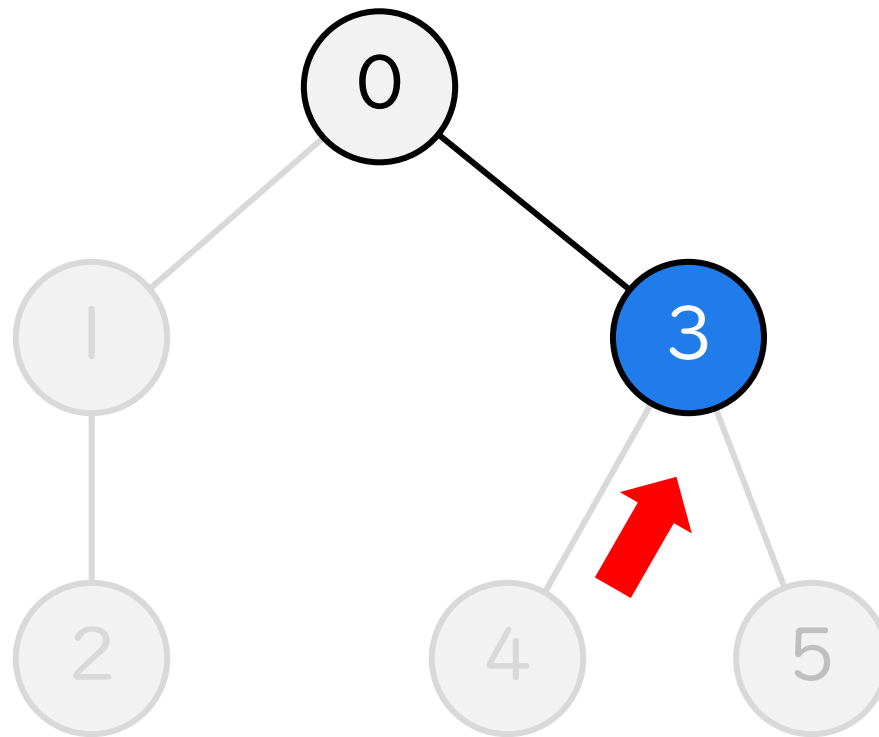
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：001100

Ali が送る方法

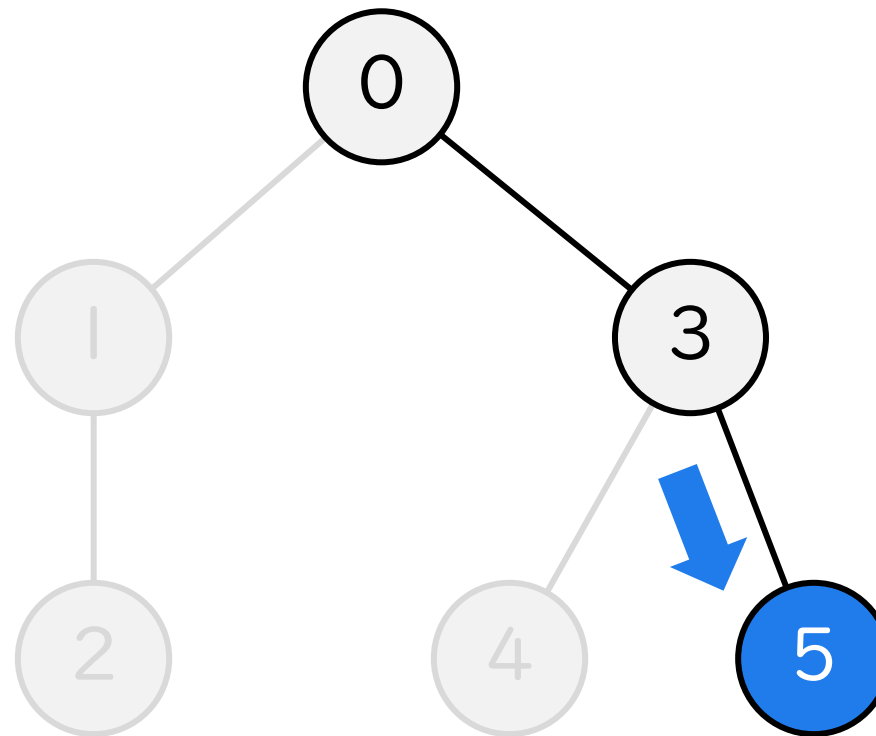
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列： 00 | 100 |

Ali が送る方法

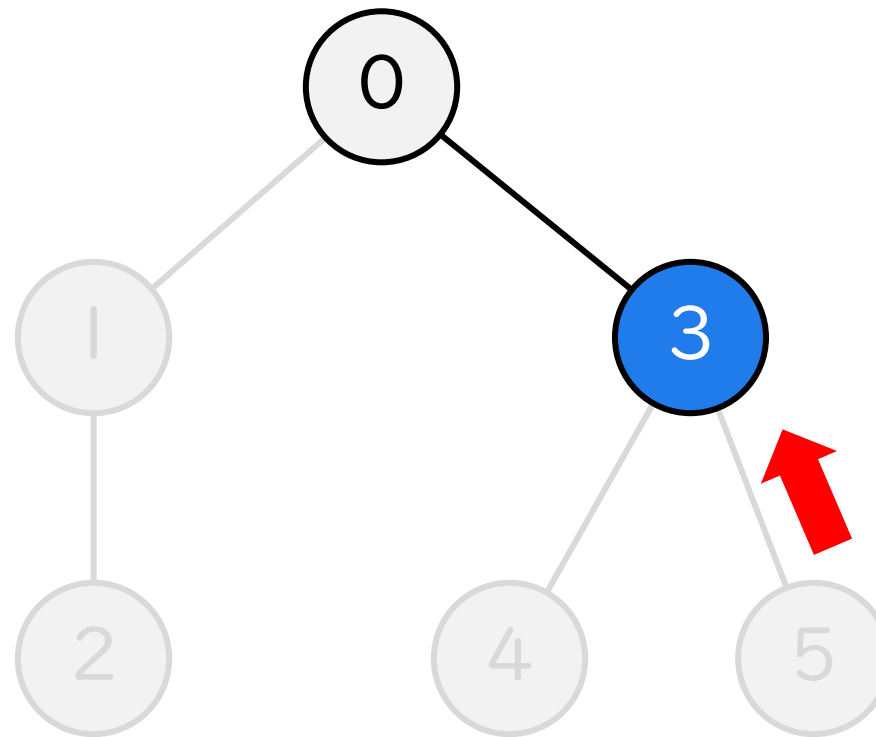
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：00110010

Ali が送る方法

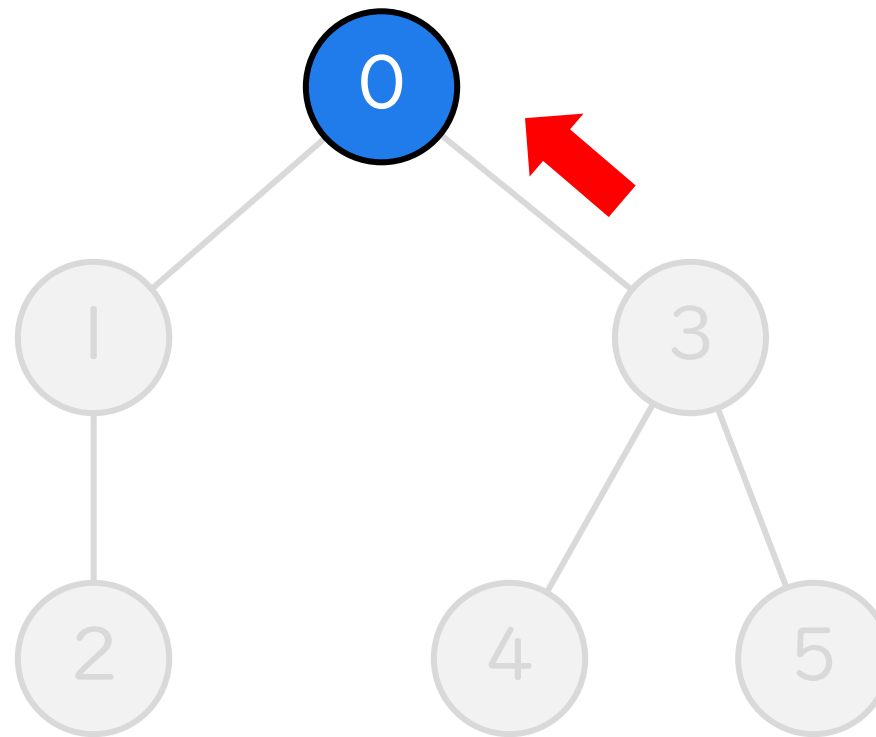
- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：001100101

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加



現在の文字列：0011001011



Ali が送る方法

- 適当な根から DFS を行う
- 進むときの

Benjamin はどうやって

木を復元するか？

現在の文字列： 00 | 100 | 0 | 1 |

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る

受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る

0

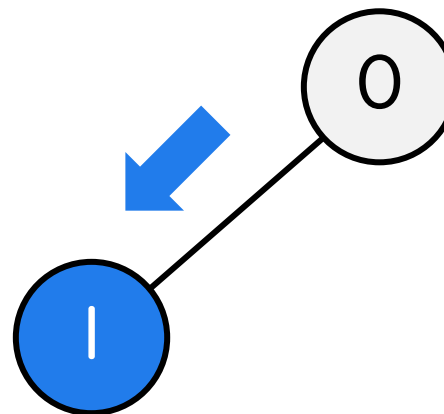
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



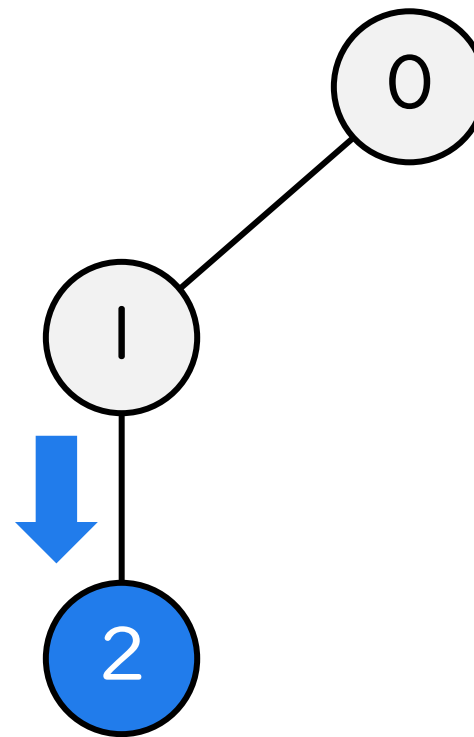
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



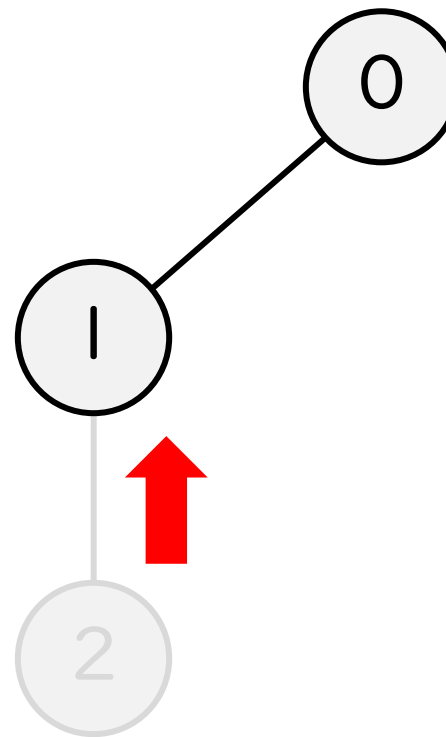
受信した文字列： **00** | | 00 | 0 | |

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



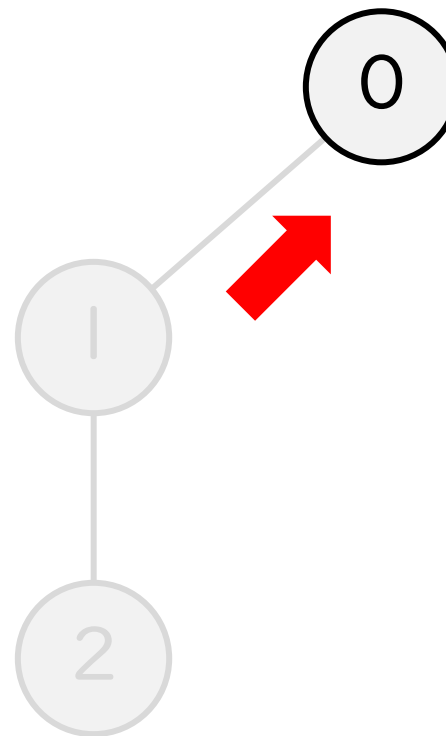
受信した文字列： 00 1 1 00 1 0 1 1

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



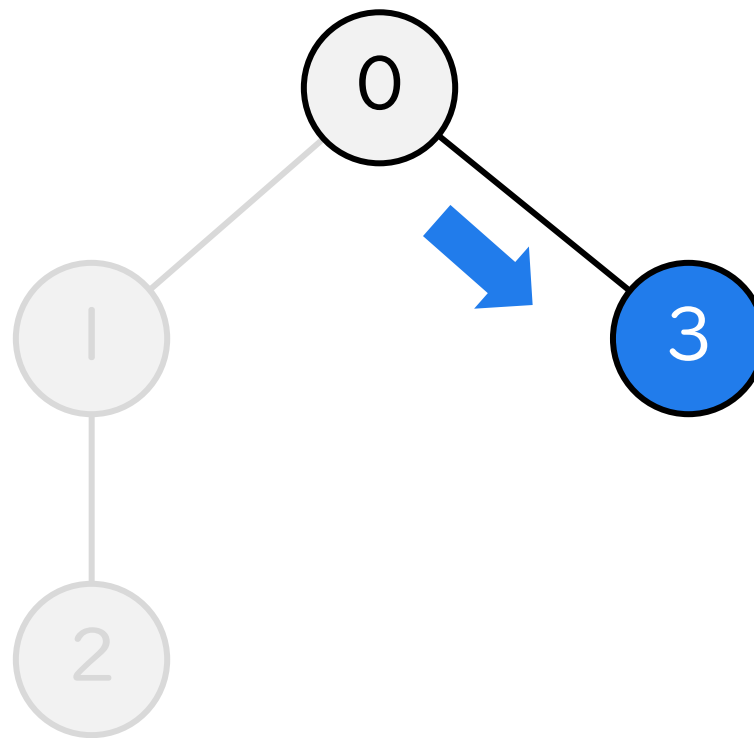
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



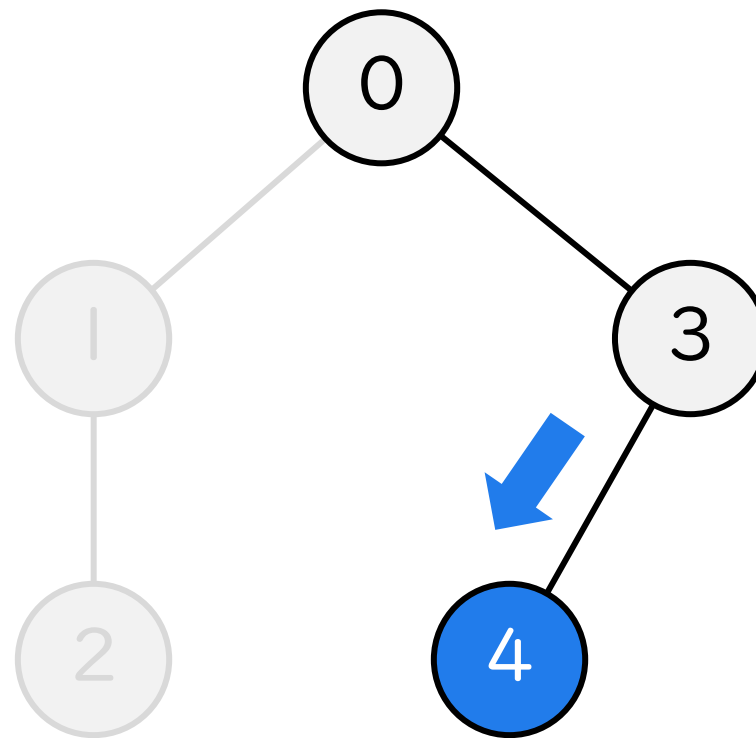
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



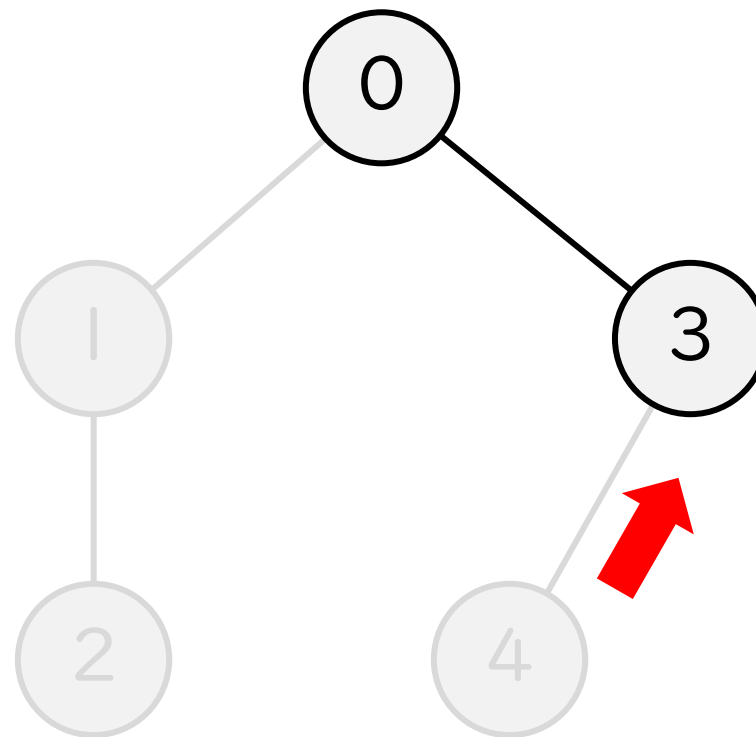
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



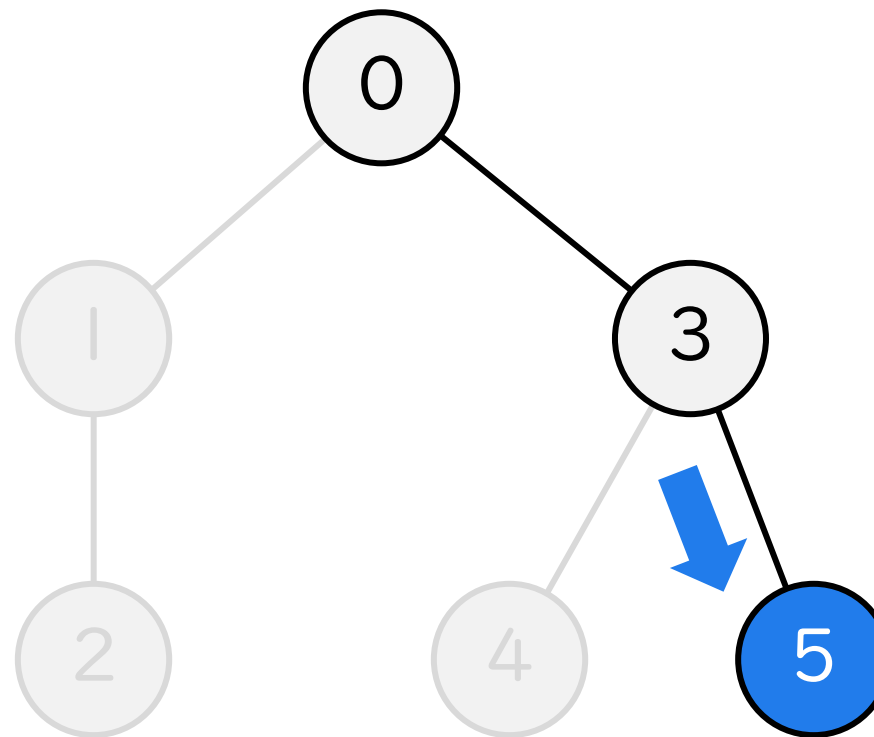
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



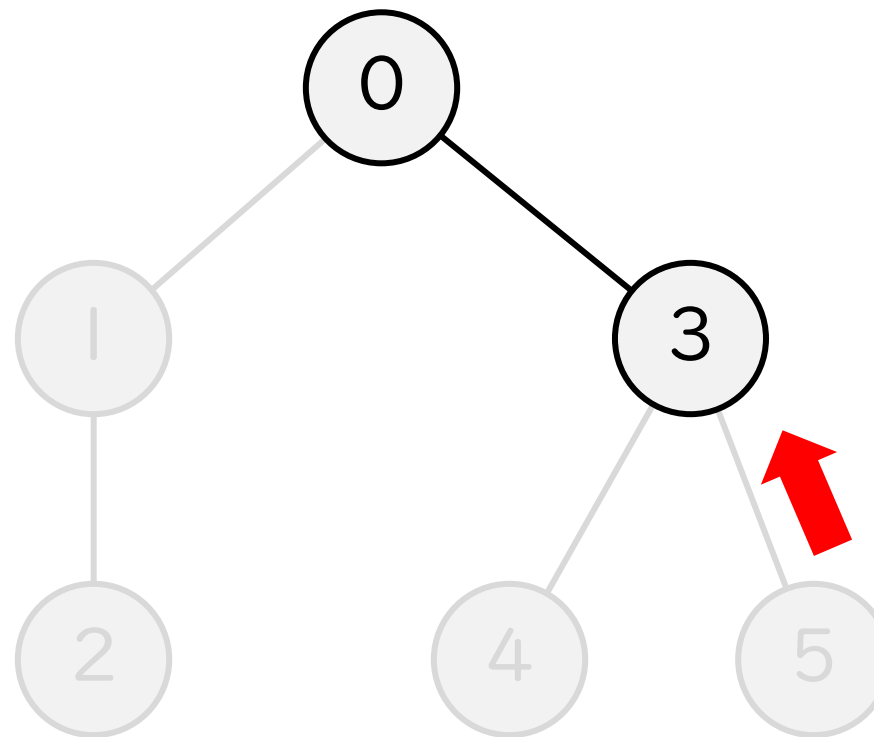
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



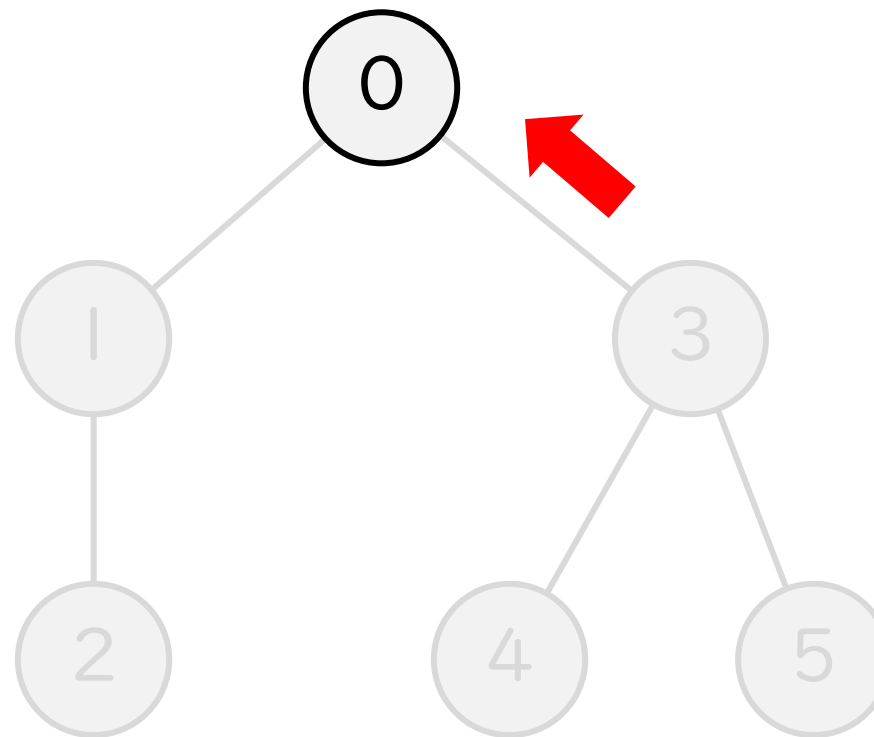
受信した文字列： 0011001011

Ali が送る方法

- 適当な根から DFS を行う
- 進むとき 0、戻るとき 1 を追加

復元する方法

- 文字列を 1 文字ずつ読んで行く
- 0 で頂点追加、1 で一步戻る



受信した文字列： 0011001011



木が正しく復元できました！

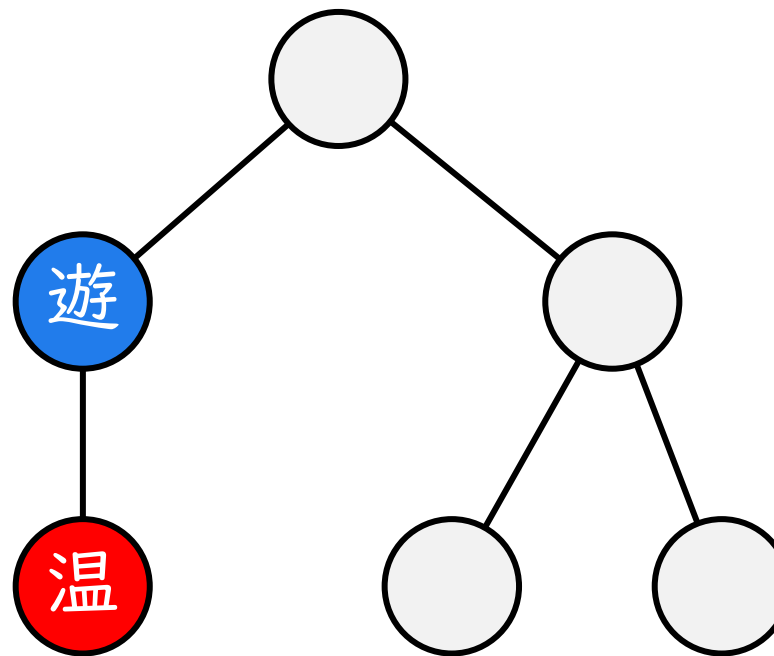
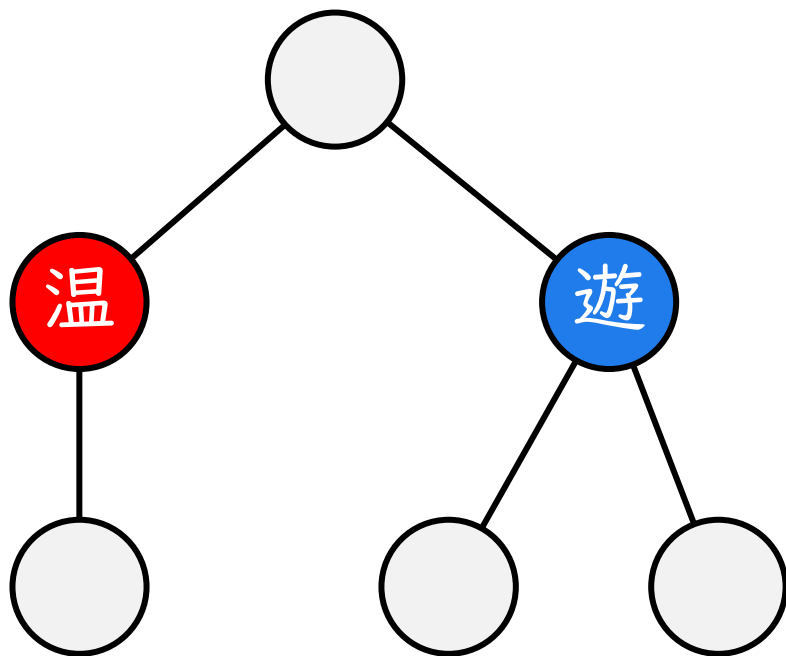
復元する方法

- 文字列を | 文字ずつ読んで行く
- 0 で頂点追加、| で一步戻る

受信した文字列： 00 | | 00 | 0 | |

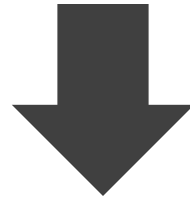
重要な問題

どの頂点が遊園地／温泉かが分からない！



重要な問題

どの頂点が遊園地／温泉かが分からない！



空港の ID を DFS の行きがけ順に
設定しよう！

必要なビット数は？

$$10000_{\text{頂点}} \times 2_{\text{ビット}} = 20000_{\text{ビット}}$$

※この解法を選んだ人は少ないかもしれませんが、解説の後半で役立ちます



必要なビット数は？

ここまでの点数： 10000 頂点 \times 2 ビット $= 15$ 点

※この解法を選んだ人は少ないかもしれませんが、解説の後半で役立ちます

Stage 2

672 ビットまでの解法

Q. Benjamin の 20 ビットを
有効に使えないか？

Benjamin が送る情報の例

頂点番号 X の下 10 ビット

頂点番号 Y の下 10 ビット

Benjamin が送る情報の例

頂点番号 X の下 10 ビット \blacktriangleright 10 通りに絞れる

頂点番号 Y の下 10 ビット \blacktriangleright 10 通りに絞れる

$$10000 \div 2^{10} \doteq 10$$

Benjamin が送る情報の例

$$10000 \div 2^{10} \doteq 10$$

Benjamin の情報によって
Ali は (X, Y) を 100 通りに絞られる

Ali が送る情報

100 通りの距離を全部、順番に送れば良い

Ali が送る情報

100 通りの距離を全部、順番に送れば良い

送るべき個数

1 個当たり

$$100 \text{ 個} \times 14 \text{ ビット} = 1400 \text{ ビット}$$

距離は 10000 まで

 $2^{14} = 16384$ より小さい



Ali が送る情報

100 通りの距離を全部、順番に送れば良い

送るべき個数

1 個当たり

ここまでの点数： 21 点

距離は 10000 まで

 $2^{14} = 16384$ より小さい

改善方法

頂点番号 X, Y をまとめて送ろう

改善方法

頂点番号 X, Y をまとめて送ろう

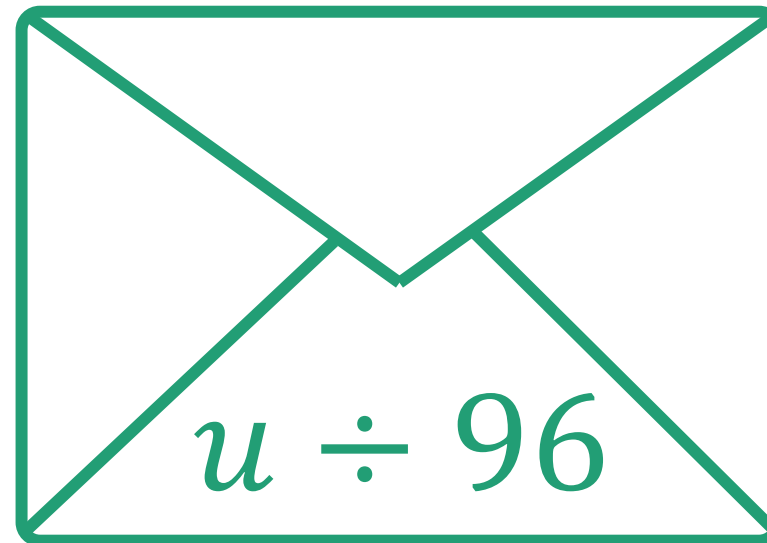
		Y の値			
		0	1	2	3
X の値	0	0	1	2	3
	1	4	5	6	7
	2	8	9	10	11
	3	12	13	14	15

各 (X, Y) に対して番号 u を定める

改善方法

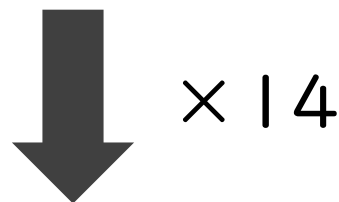
頂点番号 X, Y をまとめて送ろう

		Y の値			
		0	1	2	3
X の値	0	0	1	2	3
	1	4	5	6	7
	2	8	9	10	11
	3	12	13	14	15

各 (X, Y) に対して番号 u を定める

番号を 96 で割った値を送る

そうすると、Ali は **96** 通りしか送る必要が無くなる



最大 **1344** ビット



そうすると、Ali は 96 通りしか送る必要がなくなる

ここまでの点数： 22 点

最大 1344 ビット

遊園地 X と温泉 Y の順序は

関係ない

$X < Y$ を仮定すると？

番号の最大値が 1 億 \rightarrow 5000 万に！

		Y の値			
		0	1	2	3
X の値	0	0	1	2	3
	1	4	5	6	7
	2	8	9	10	11
	3	12	13	14	15

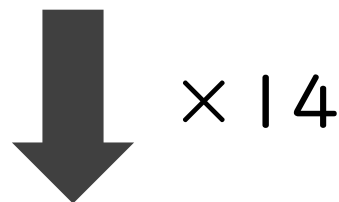
$X < Y$ を仮定しない



		Y の値			
		0	1	2	3
X の値	0		0	1	2
	1			3	4
	2				5
	3				

$X < Y$ を仮定する

そうすると、Ali は **48** 通りしか送る必要が無くなる



$\times 14$

最大 **672** ビット



そうすると、Ali は 48 通りしか送る必要がなくなる

ここまでの点数 $\times 4$: 32 点

最大 672 ビット

※ここまでの方針とオイラーツアーを組み合わせると 50 点台まで狙えますが

満点解法とは関係ないので解説では詳しく扱いません

※なぜ 48 か？ → 番号の最大値は 5000 万であり、48 で割ればギリギリ 2^{20} に収まるため

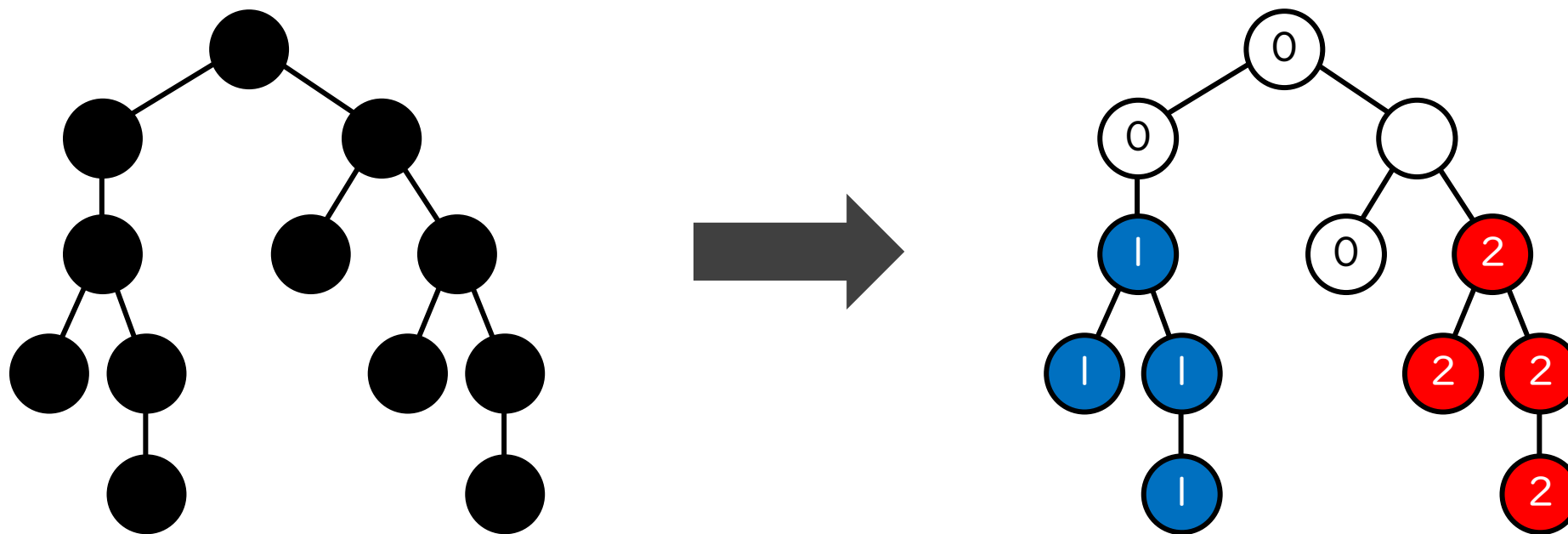
Stage 3

118 ビットまでの解法

本質的な考察

頂点数の差が 2 倍以内になるように
木を連結なグループに分解することができる

※この問題の制約の場合。また、1つのグループを除く。

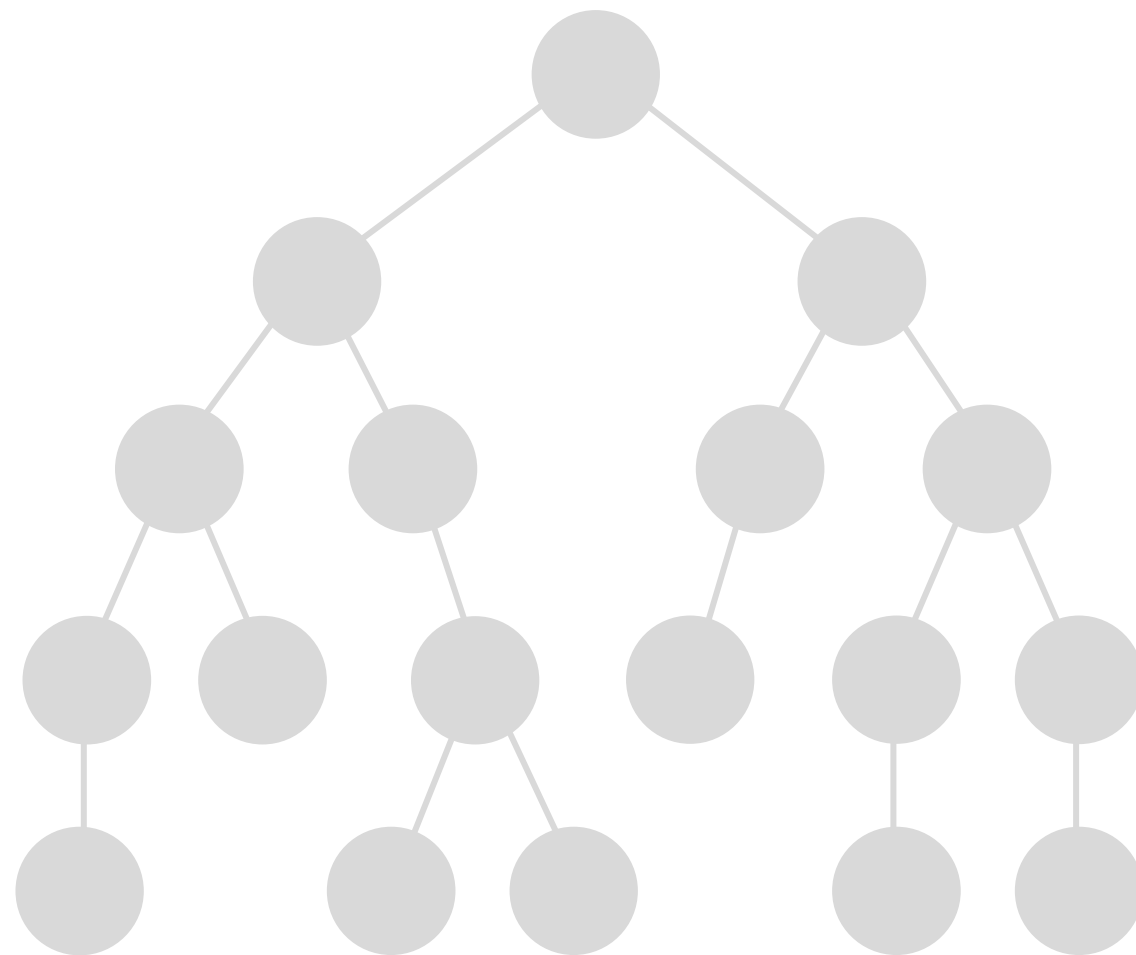


木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る

木の分解方法

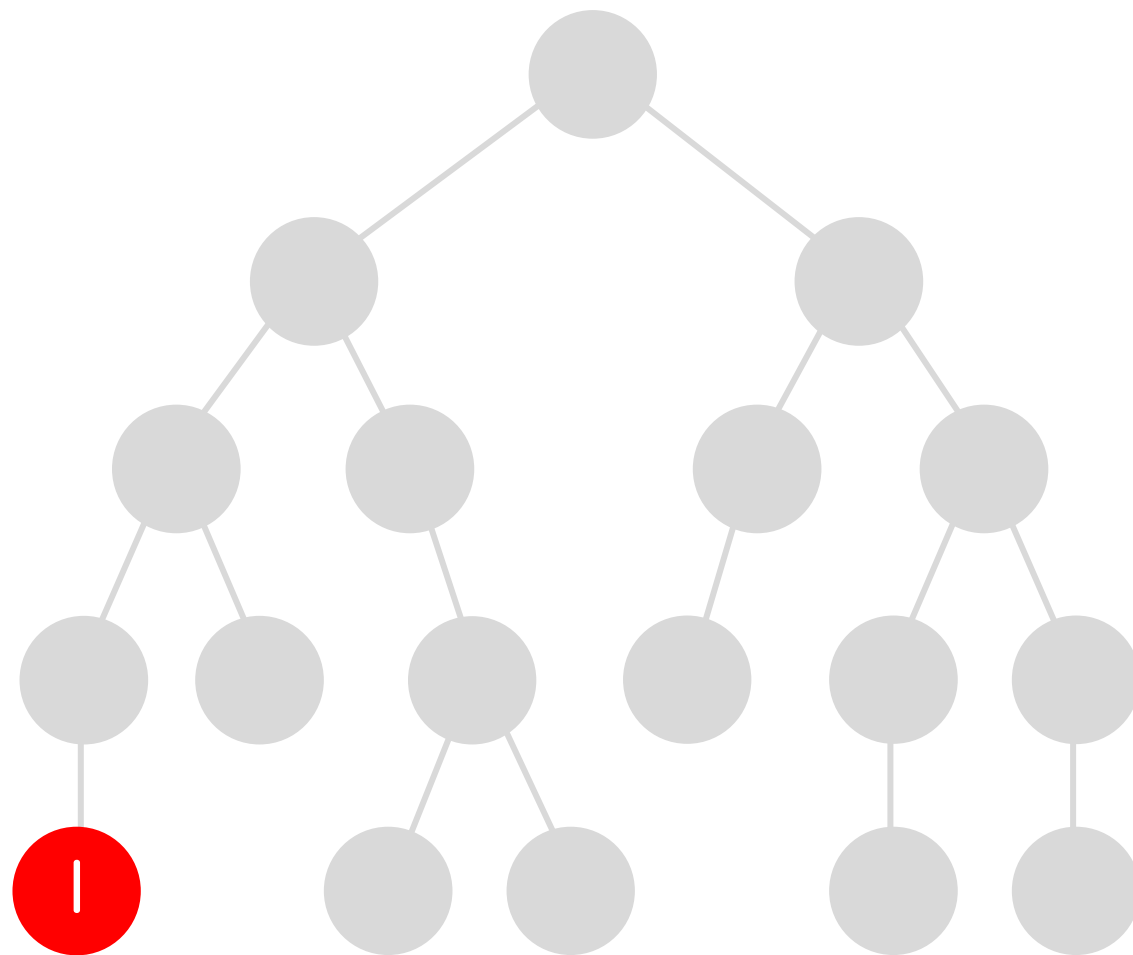
- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

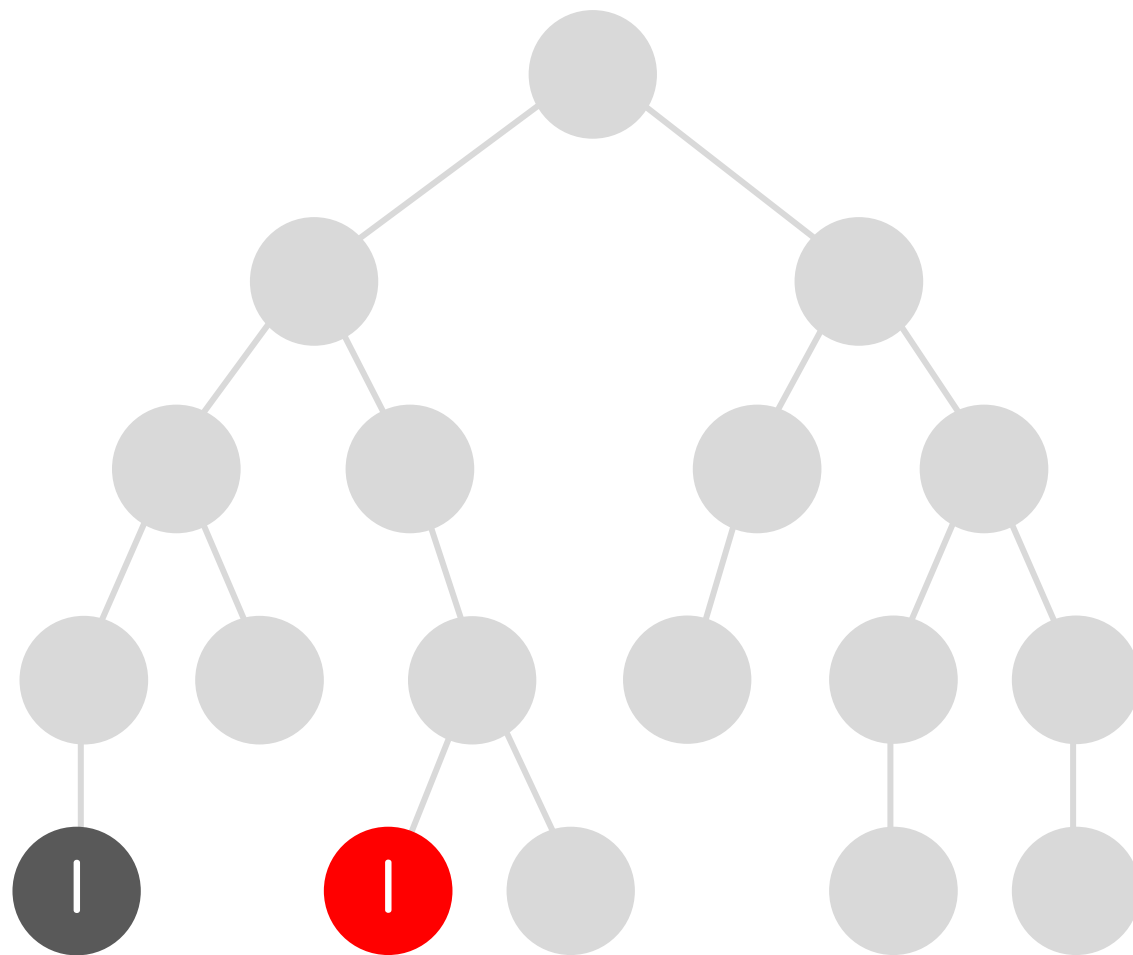
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

※頂点の数字は
現在のグループのサイズ

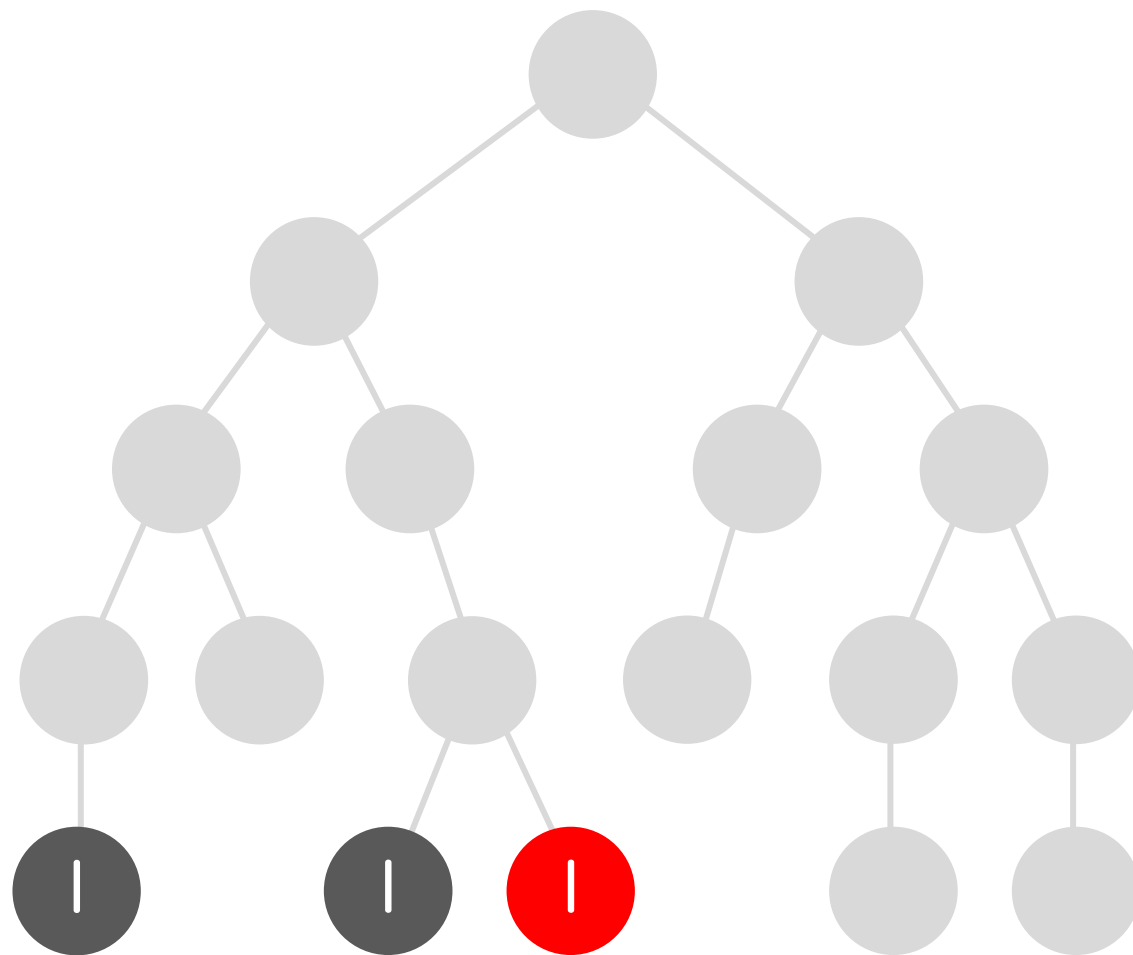


木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

※頂点の数字は

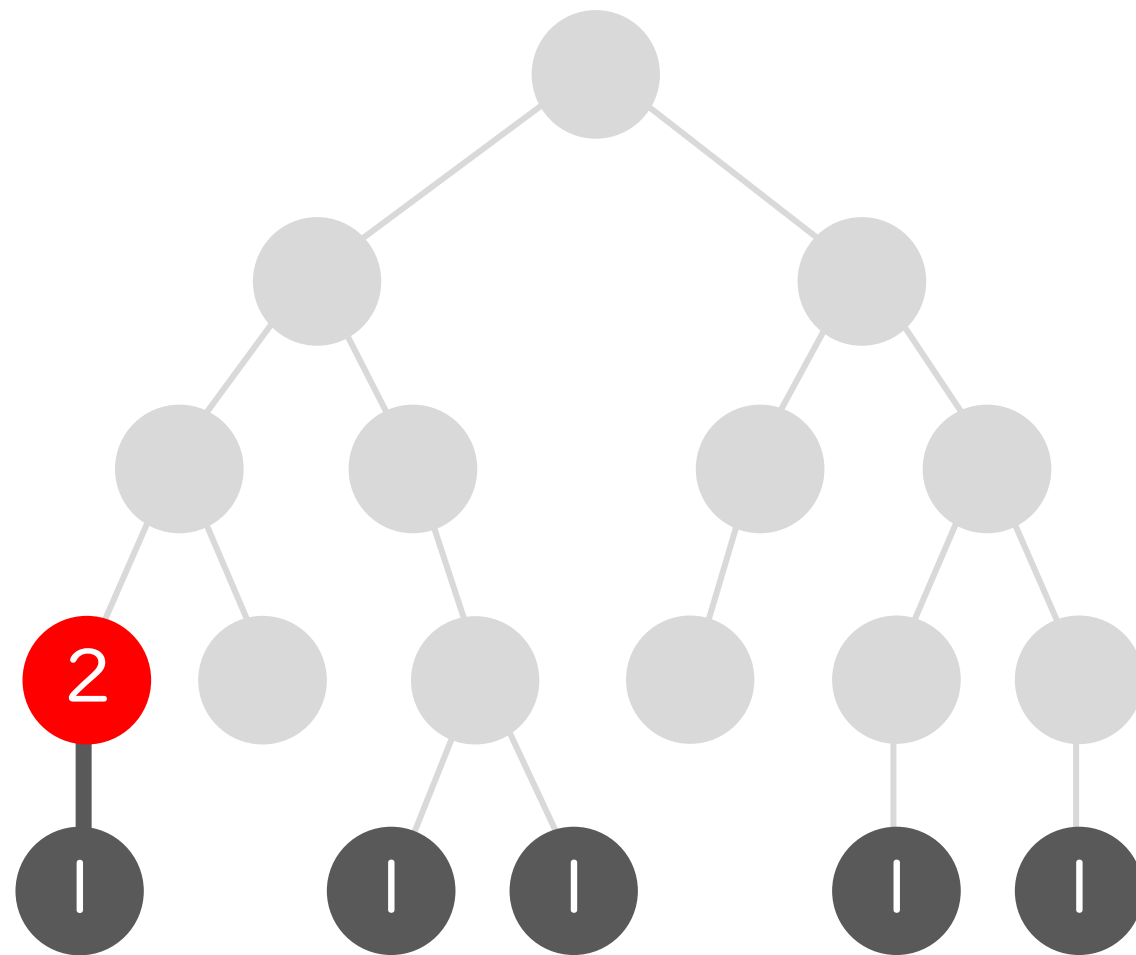
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

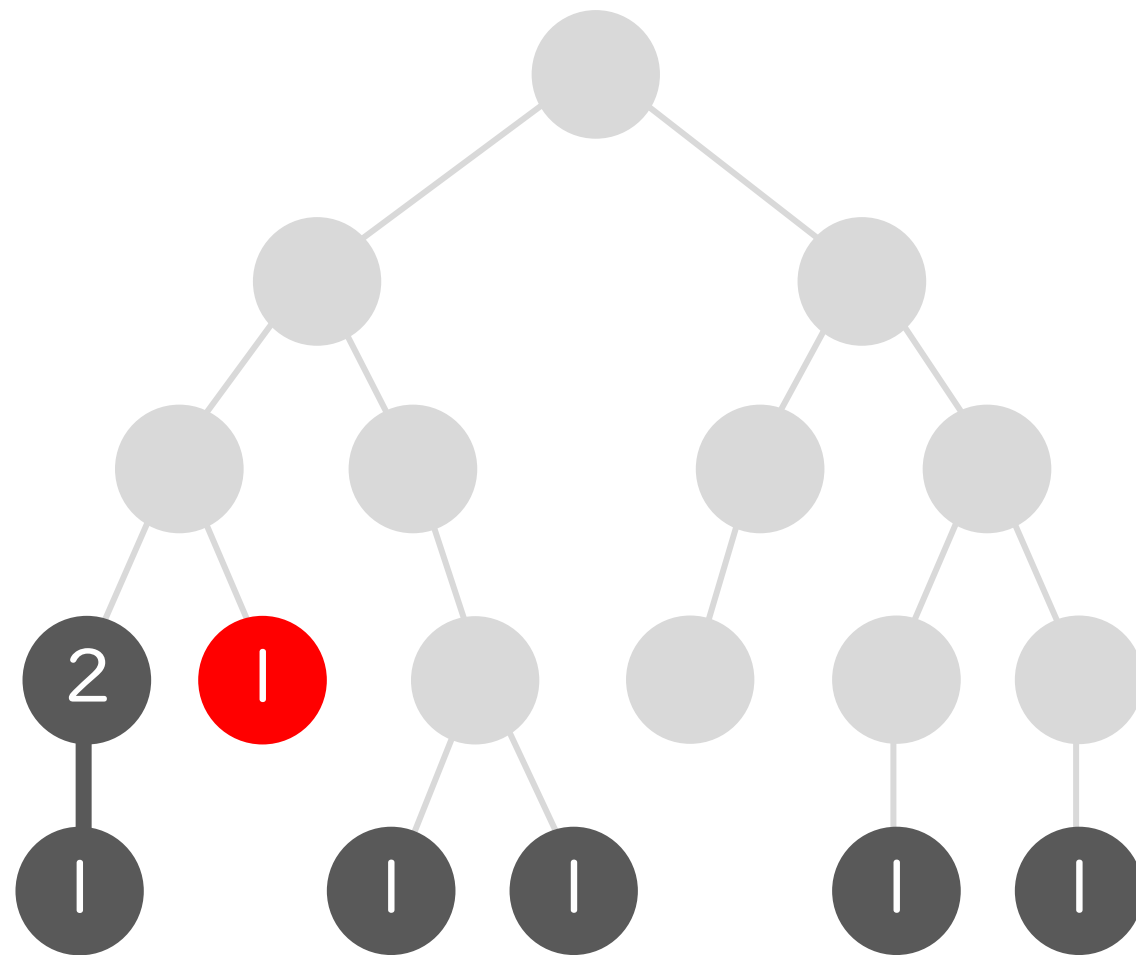
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

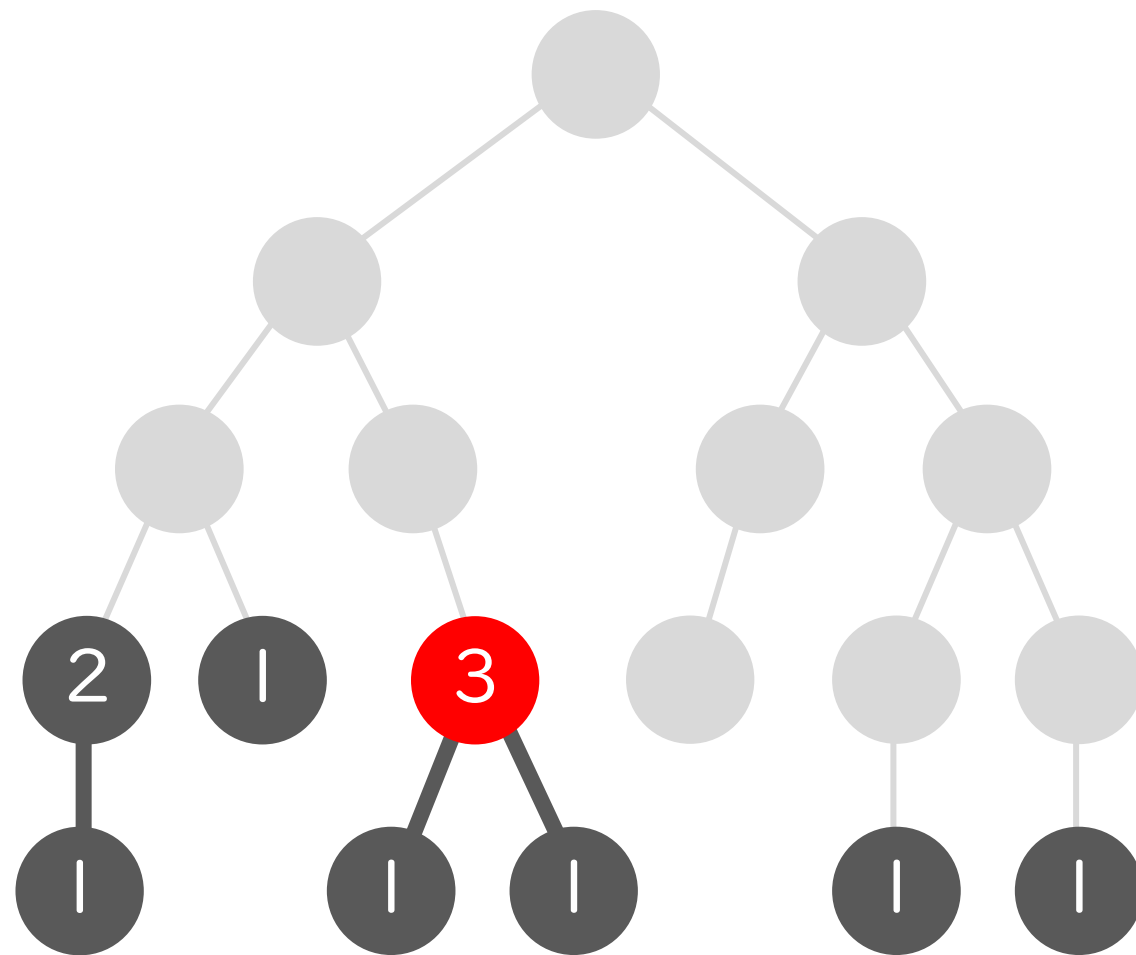
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

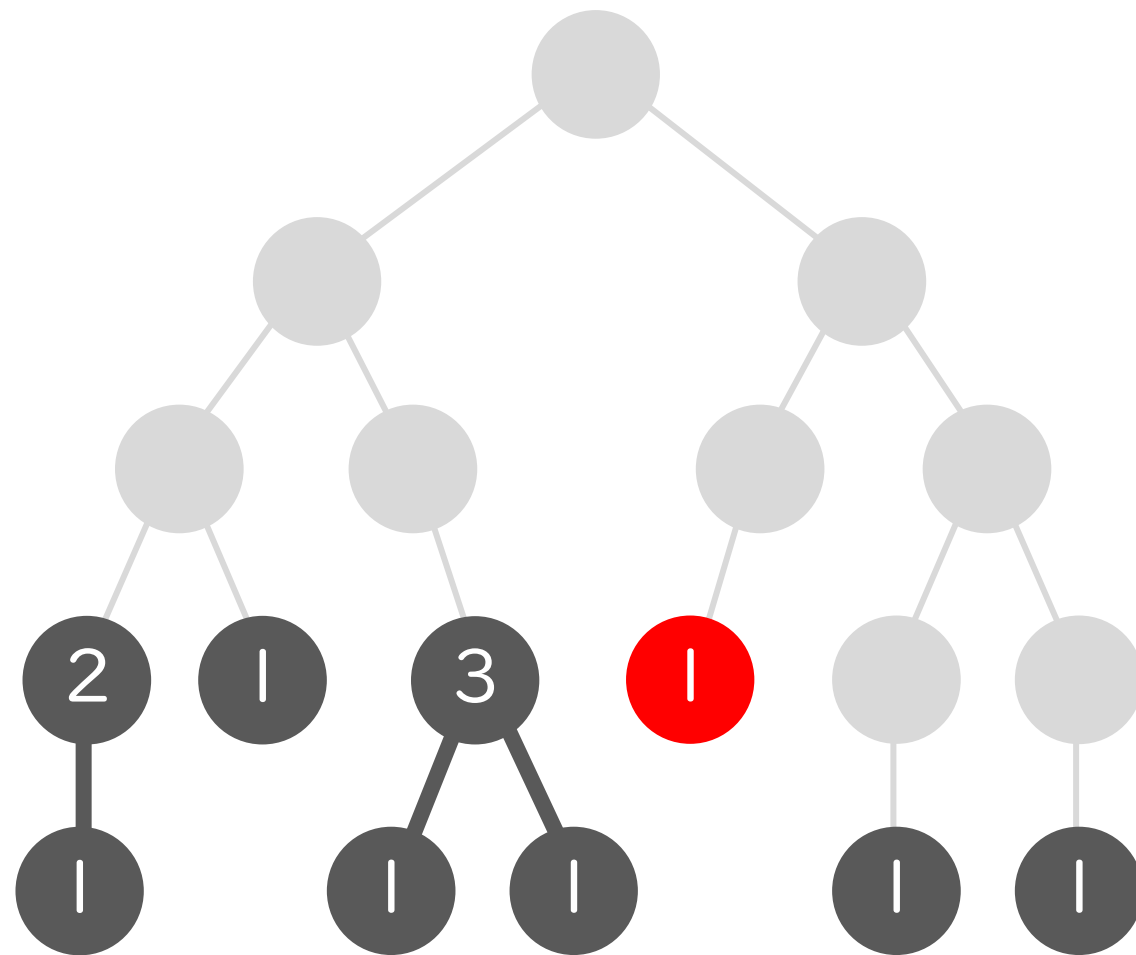
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

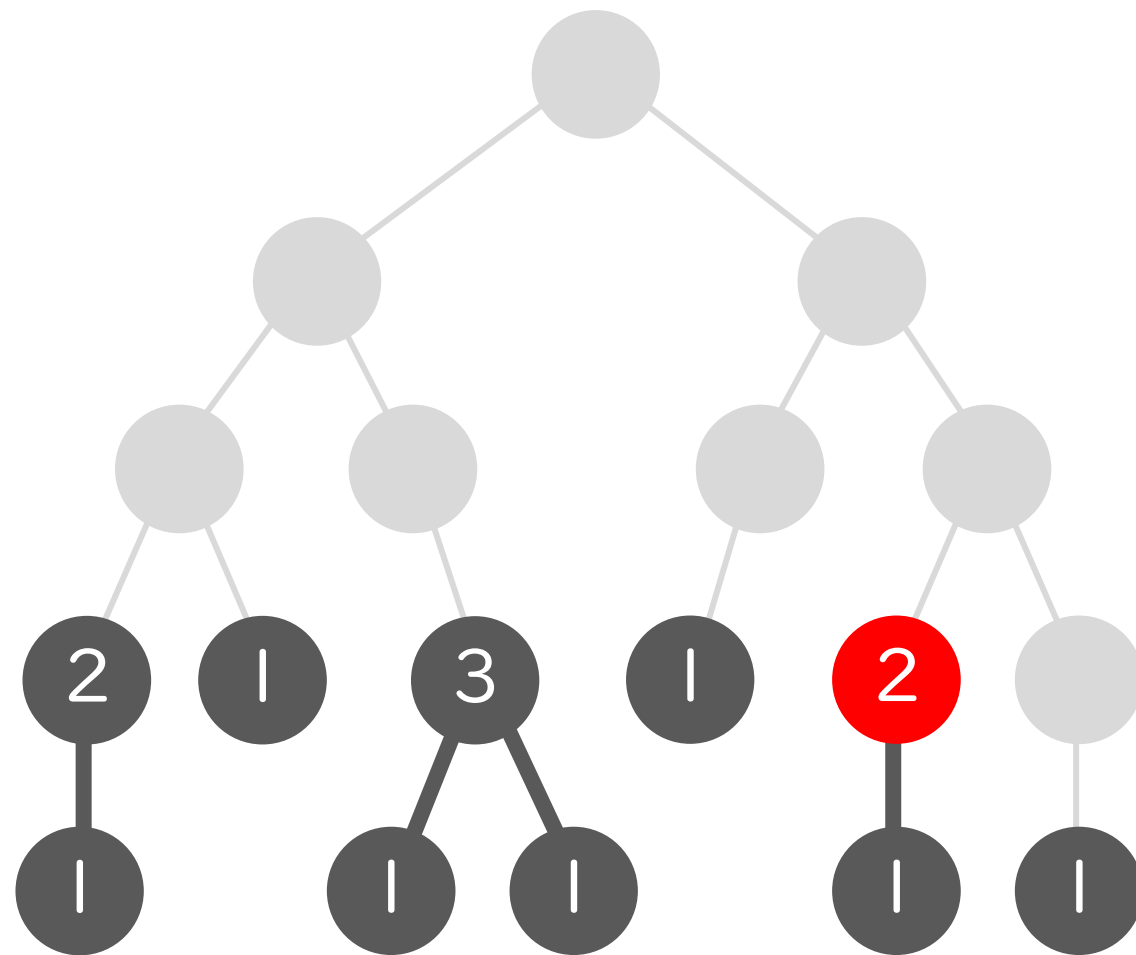
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

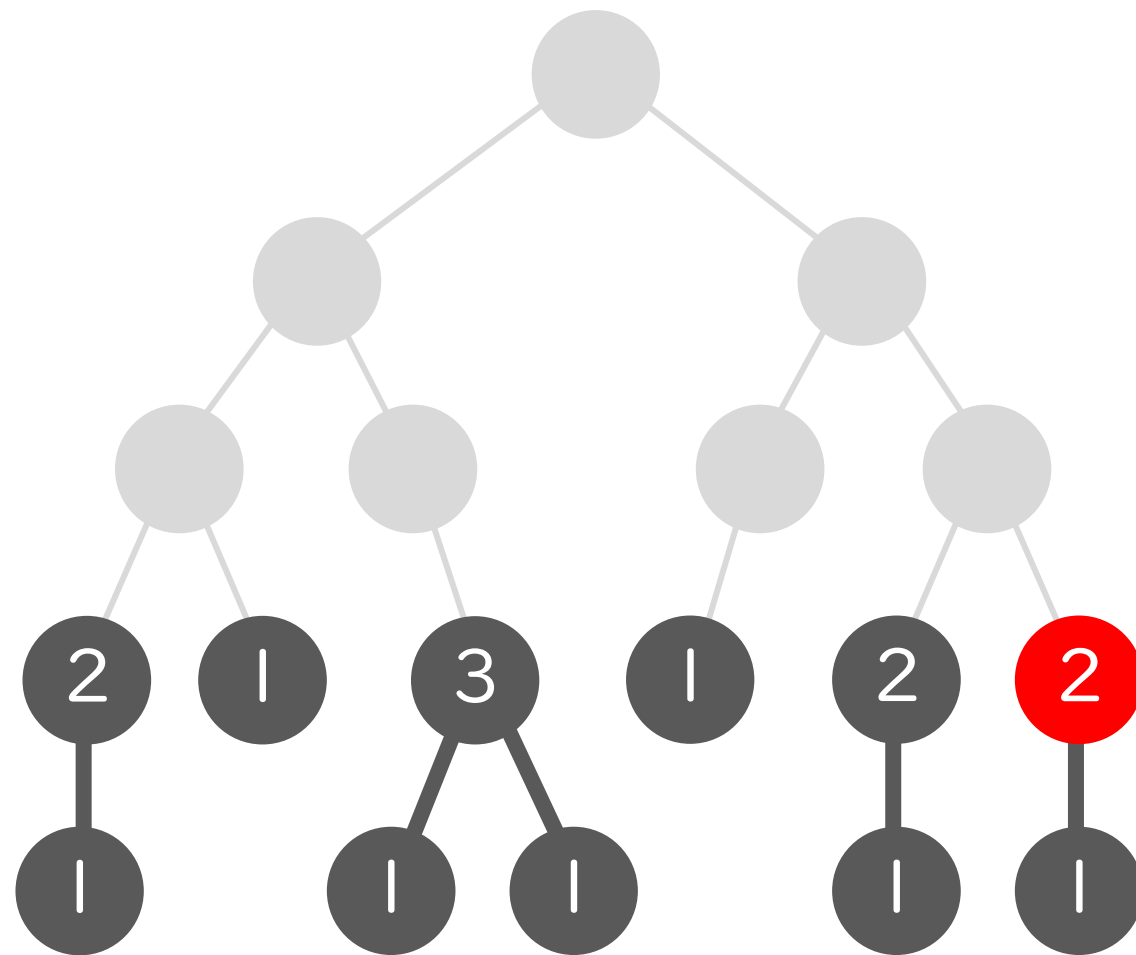
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

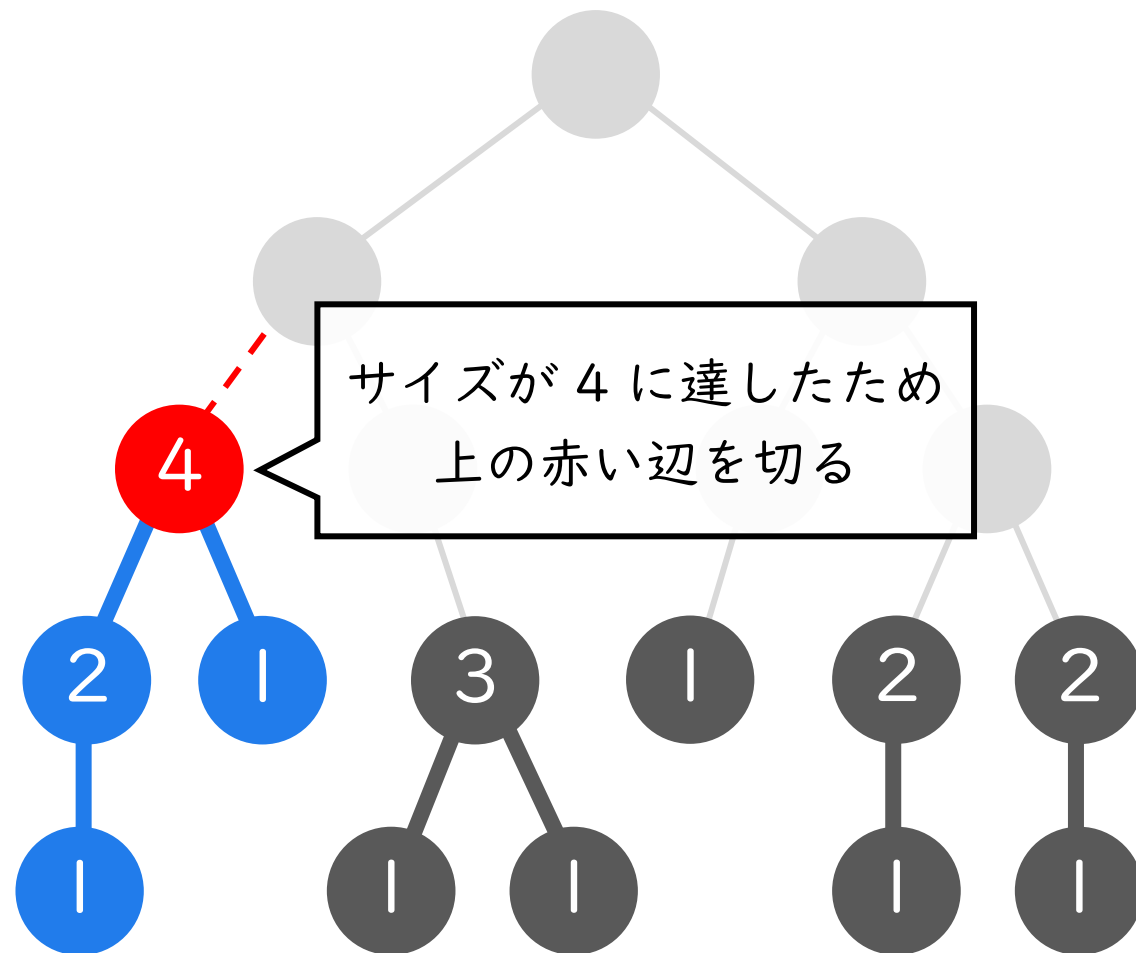
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

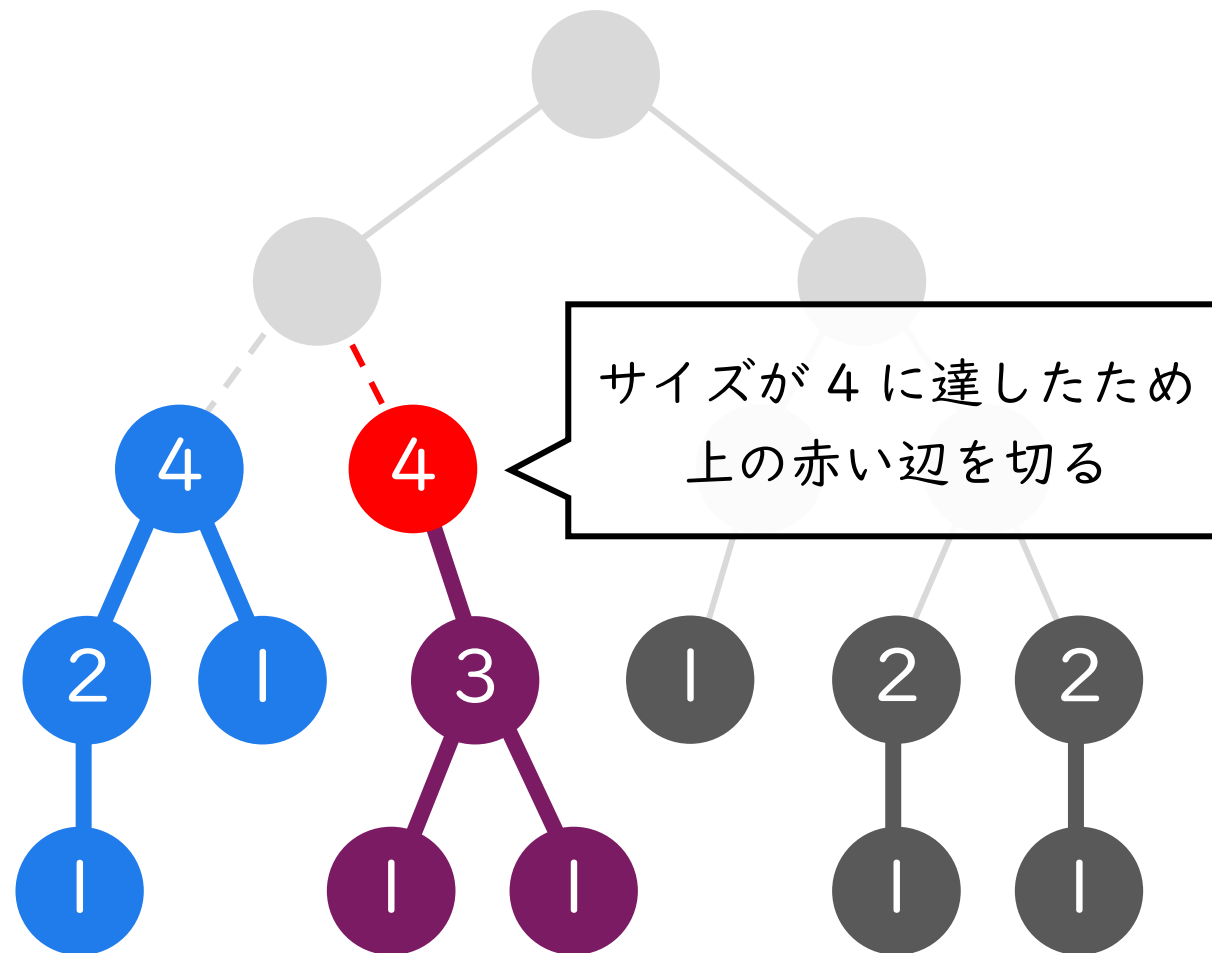
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

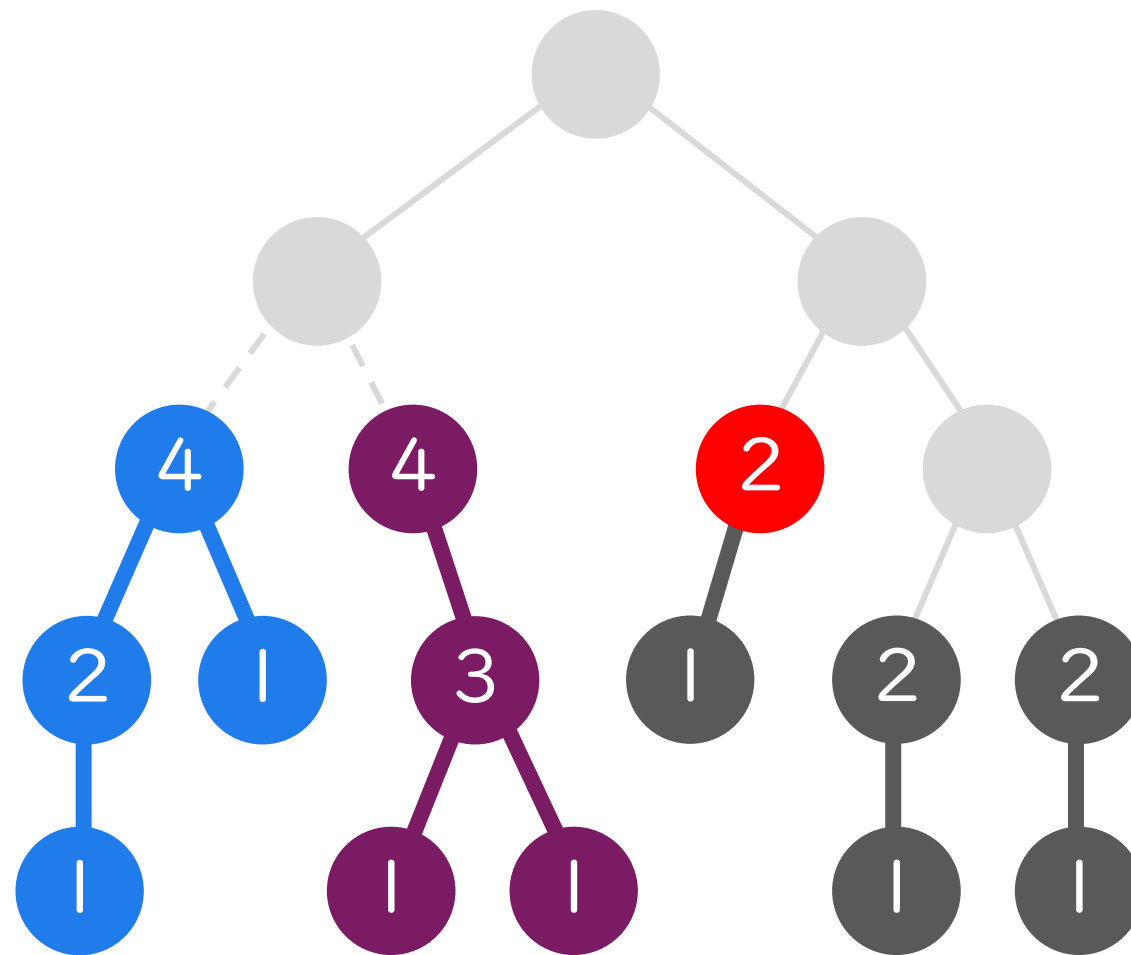
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

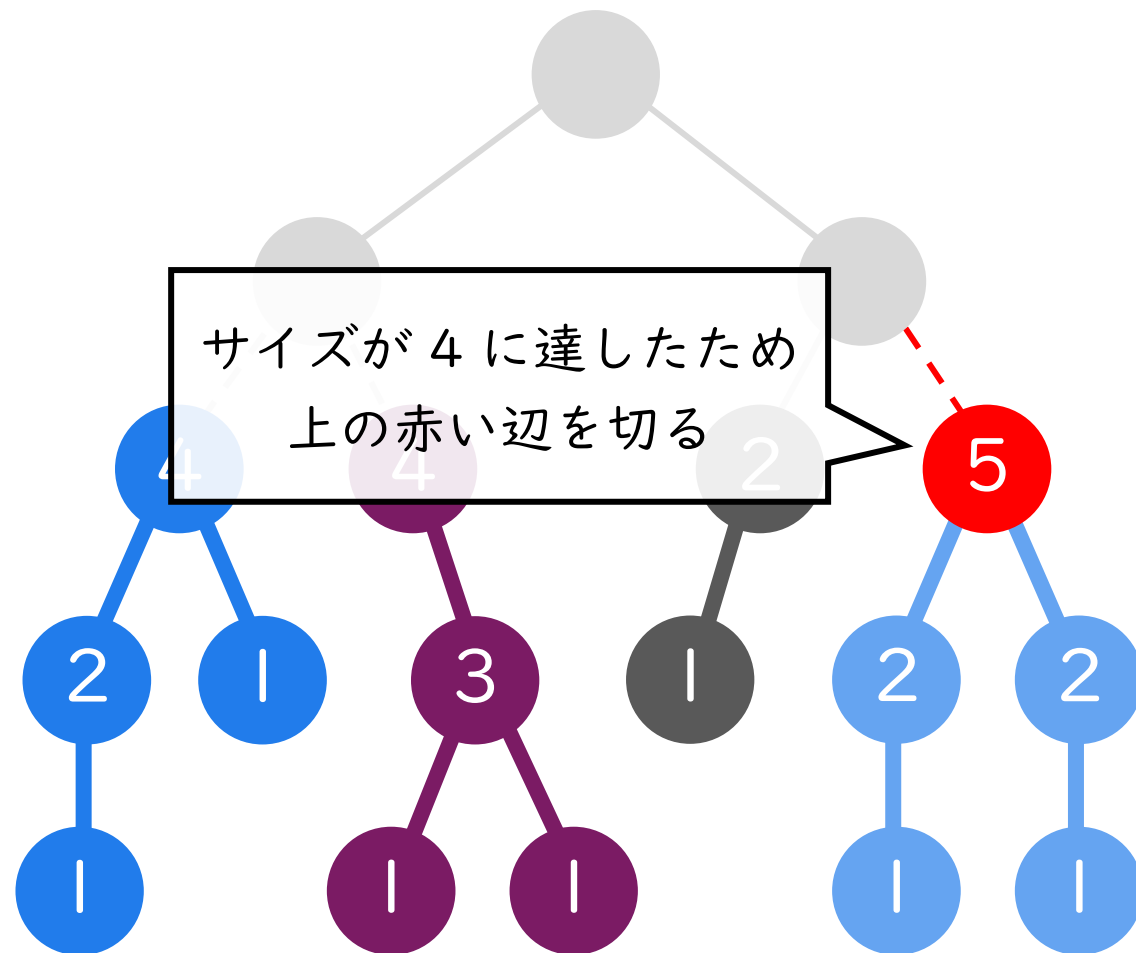
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

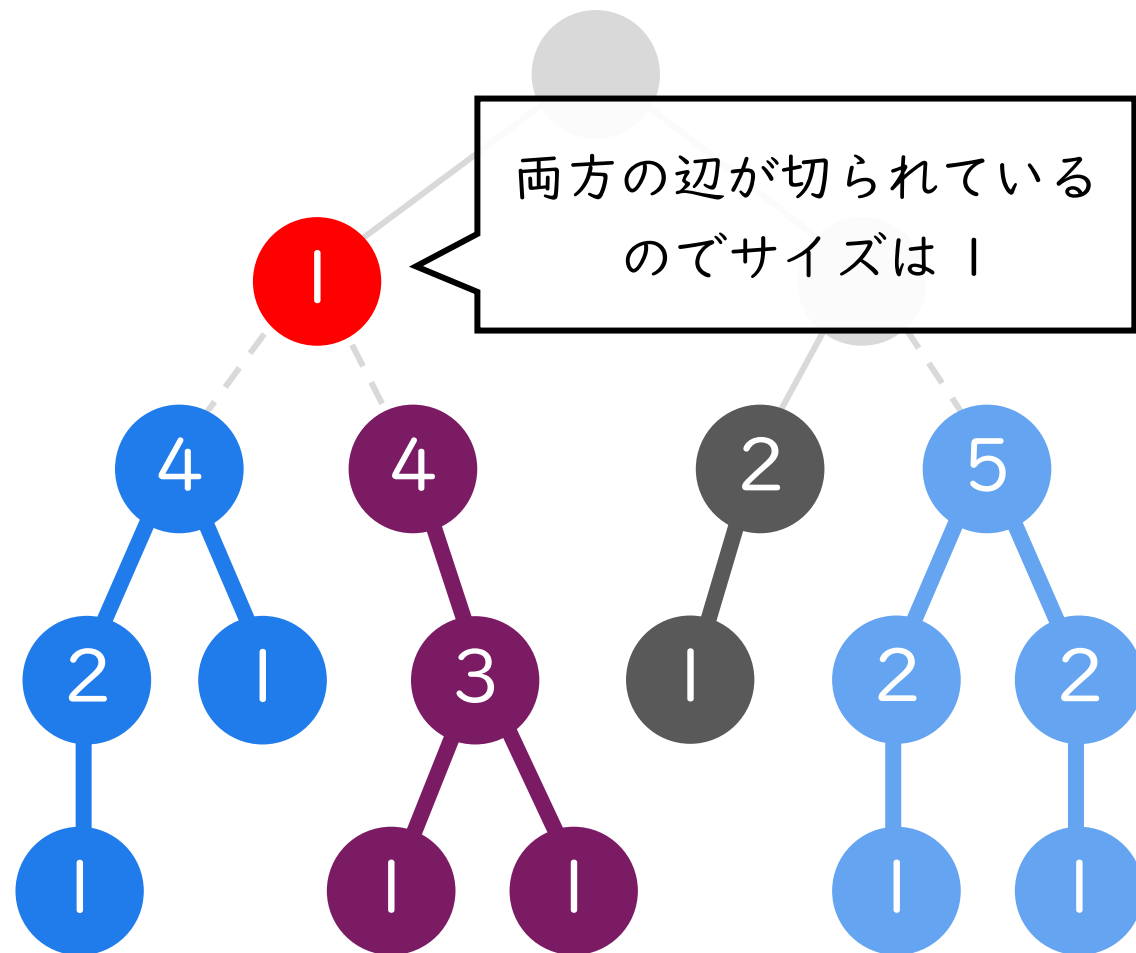
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

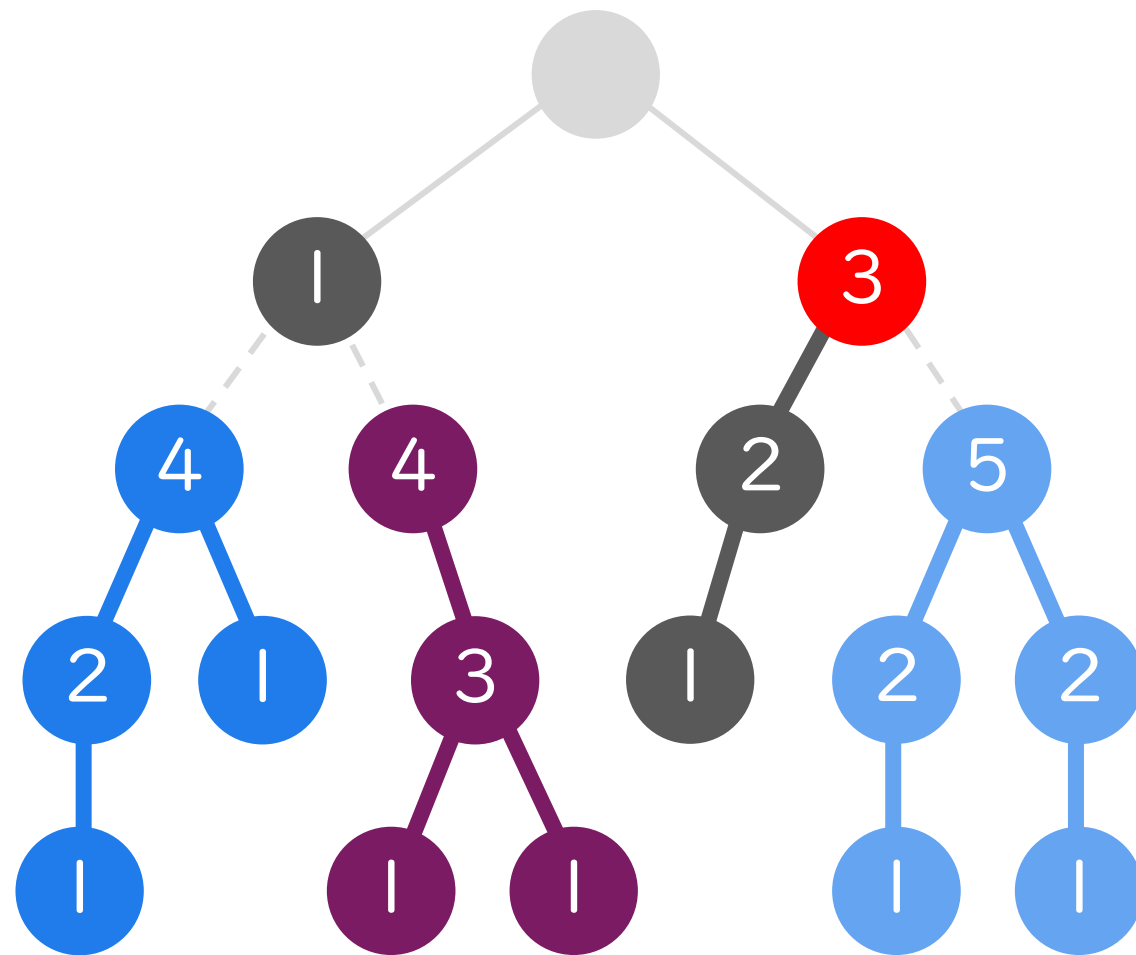
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

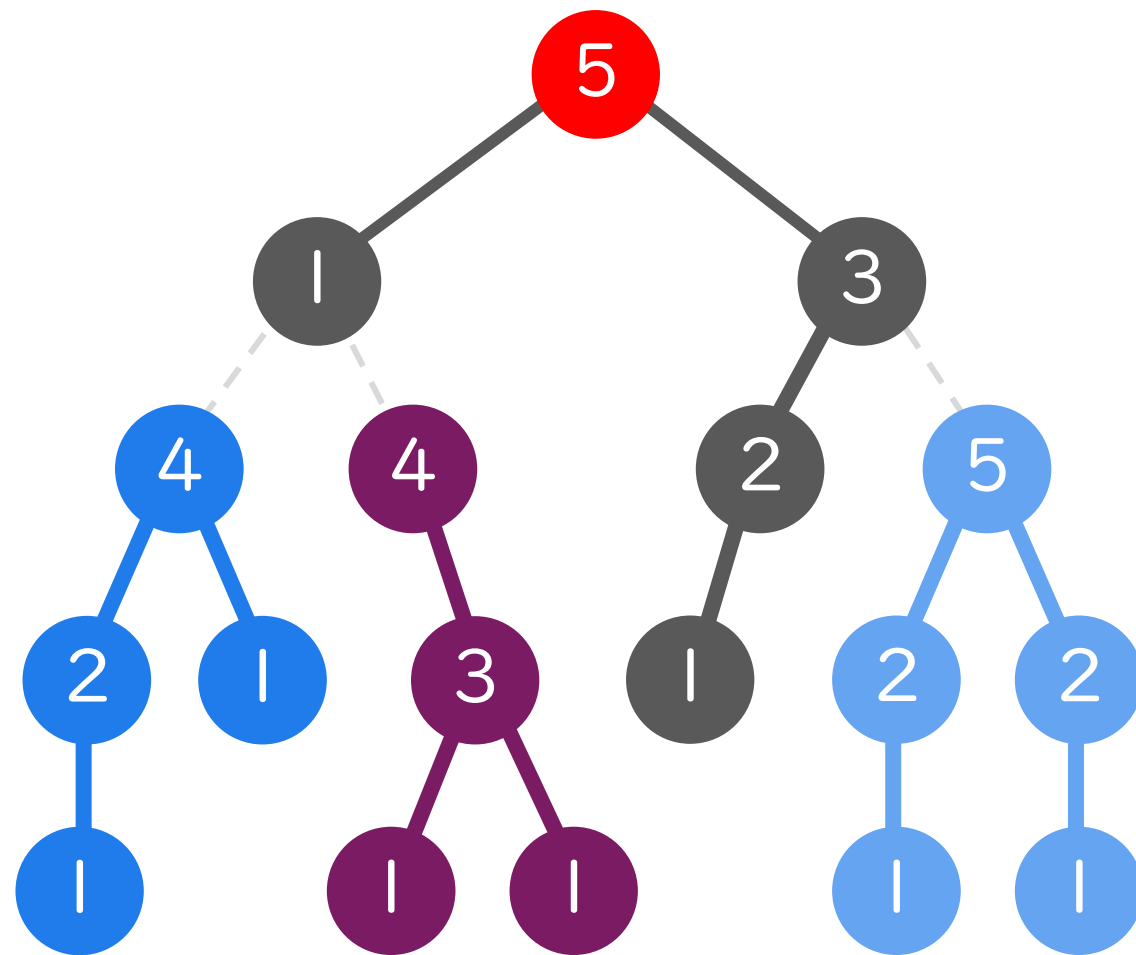
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

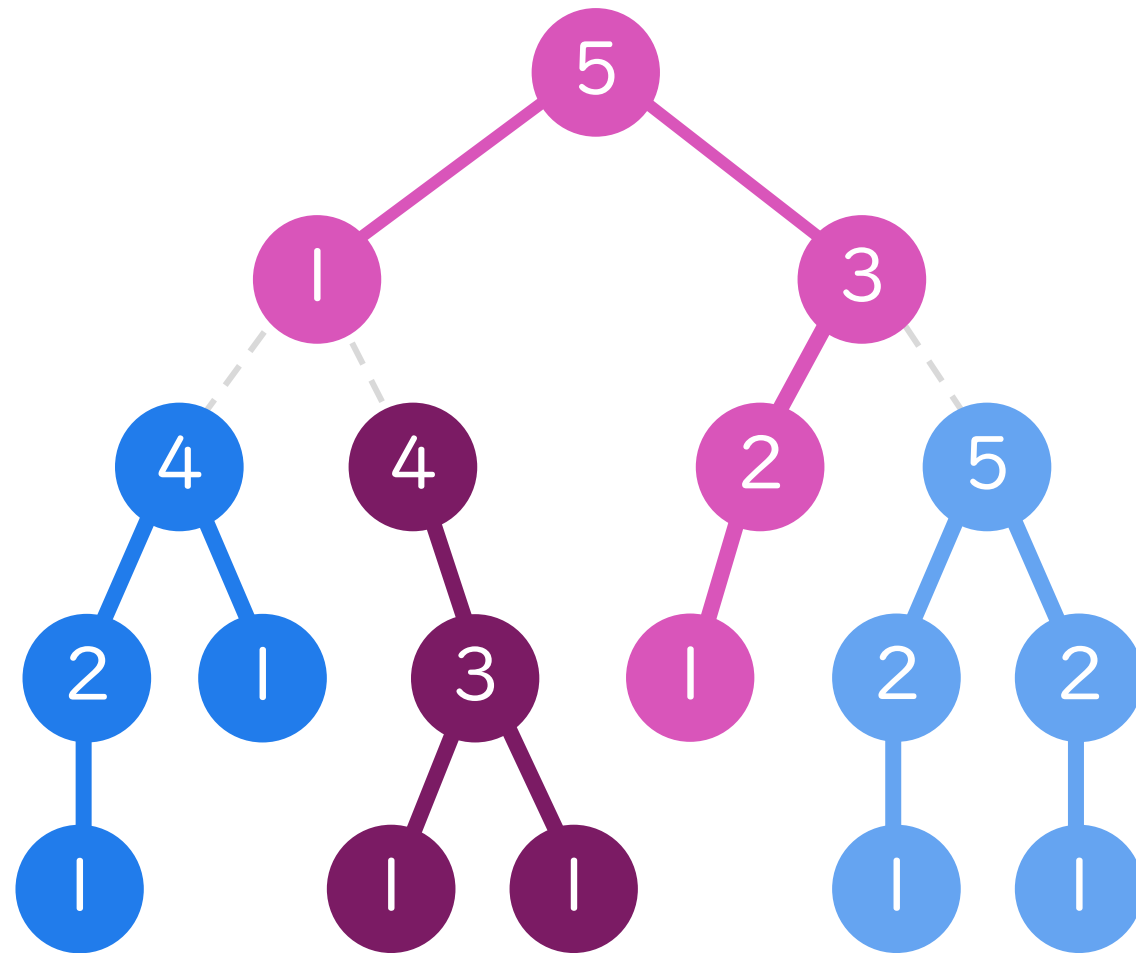
※頂点の数字は
現在のグループのサイズ



木の分解方法

- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

※頂点の数字は
現在のグループのサイズ



木の分解方法

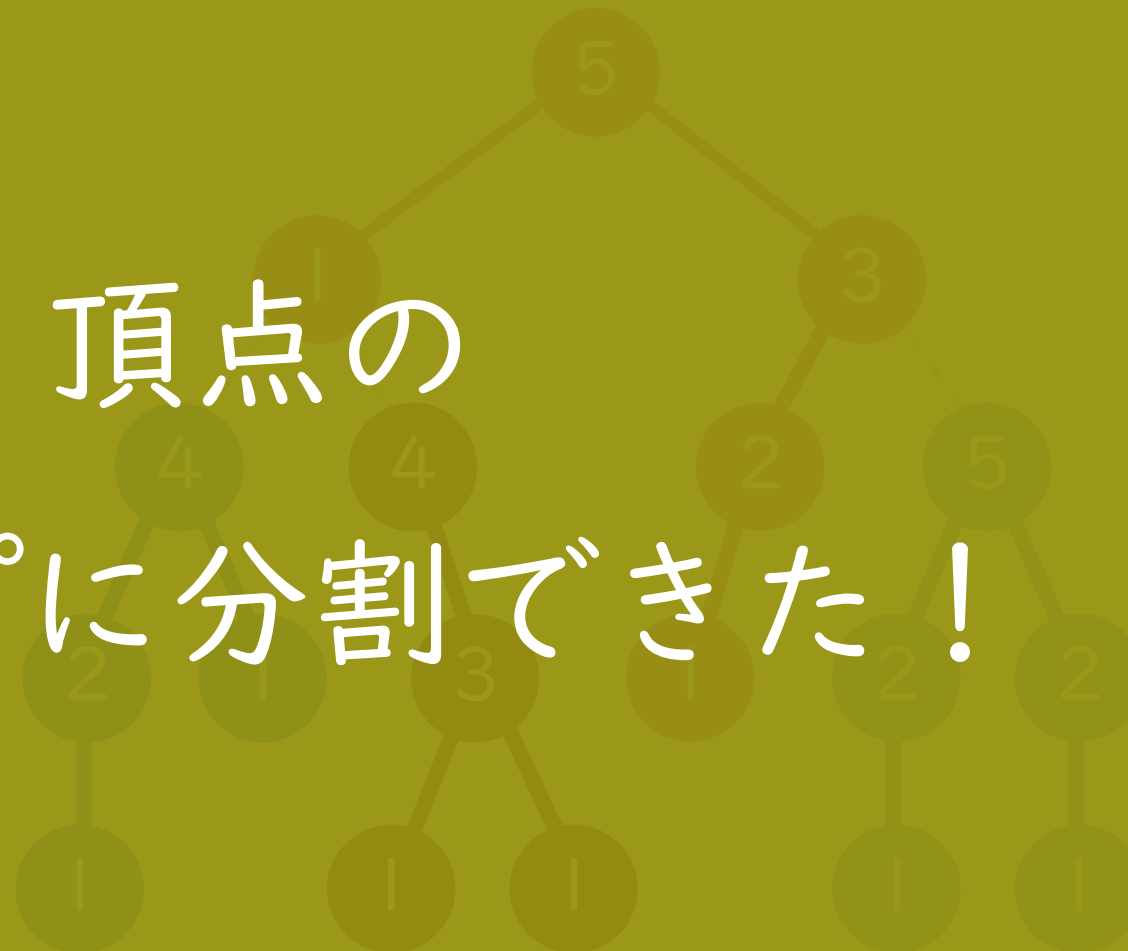
- 下から貪欲に分解していく
- グループのサイズが定数 B を超えた場合、辺を切る
- 右図は $B = 4$ の例

最大 5 頂点の

4 つのグループに分割できた！

※頂点の数字は

現在のグループのサイズ



Q. この方法を使うと

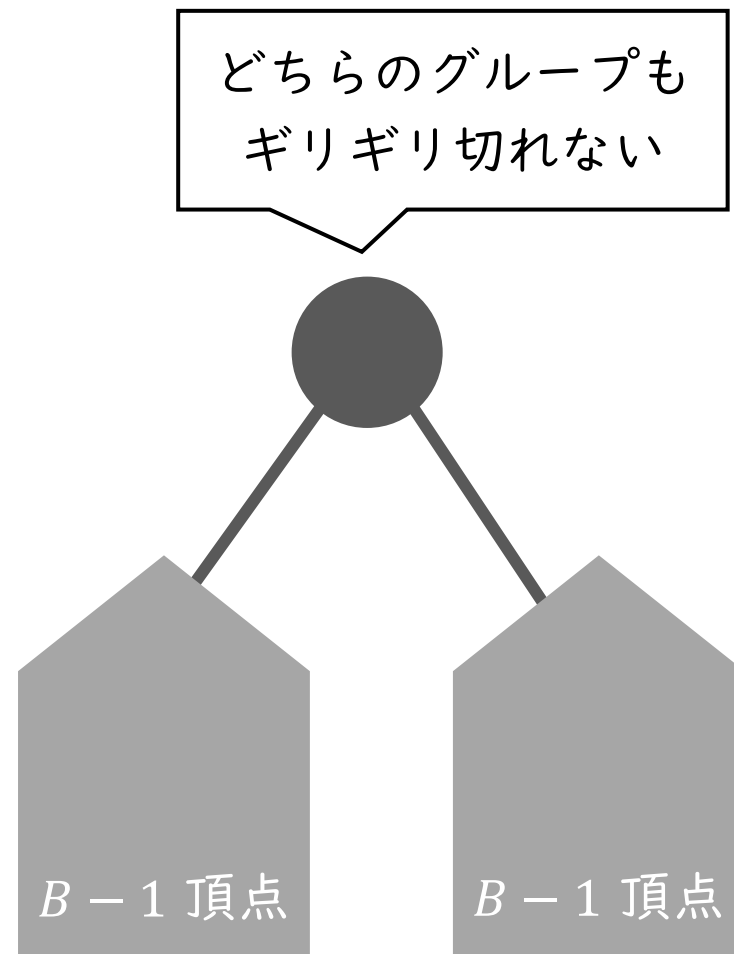
最も大きいグループの頂点数はどうか？

次数は 3 以下なので…

根付き木の次数は **2 以下**

最悪ケースは右図の通り

最大 **$2B-1$** 頂点



次数は 3 以下なので…

どちらのグループも
ギリギリ切れない

最大サイズ $2B - 1$ 以下の

高々 $\left\lfloor \frac{N}{B} \right\rfloor$ 個のグループに分割可能

最大 $2B - 1$ 頂点

$B - 1$ 頂点

$B - 1$ 頂点

※根を含むグループ以外は、頂点数 B 以上 $2B - 1$ 以下

たとえば最小サイズ $B=7$ で木をグループ分けしたとする



どのような戦略で通信するか？

ステップ 1

空港の ID の設定

空港の ID の設定

$14 \times (\text{グループ番号})$
 $+ (\text{グループ内頂点番号})$

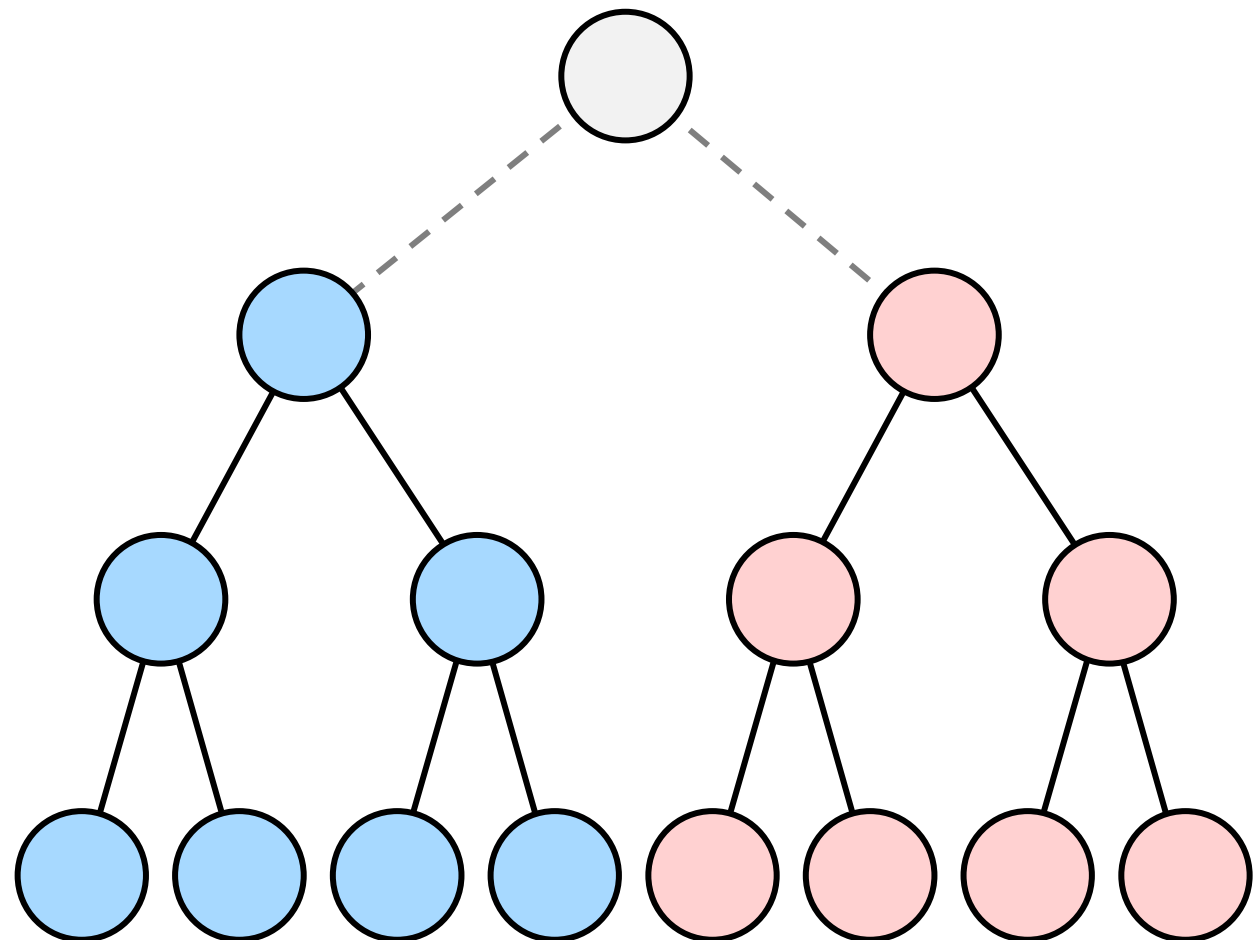
※グループ内頂点番号は適当に割り振る

1 グループの
頂点数は 13 以下

空港の ID の設定

14 × (グループ番号)
+ (グループ内頂点番号)

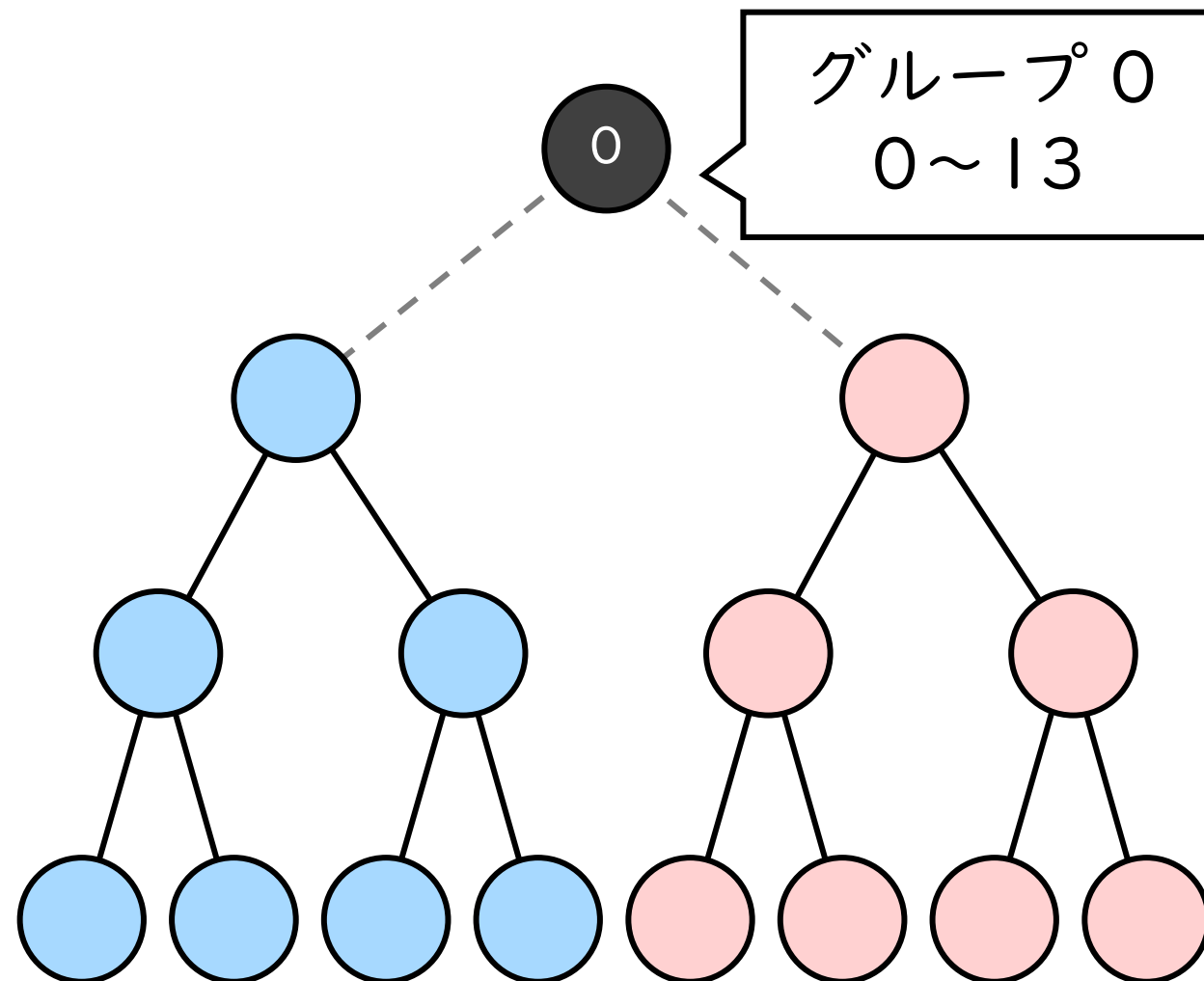
※グループ内頂点番号は適当に割り振る



空港の ID の設定

$14 \times (\text{グループ番号})$
 $+ (\text{グループ内頂点番号})$

※グループ内頂点番号は適当に割り振る

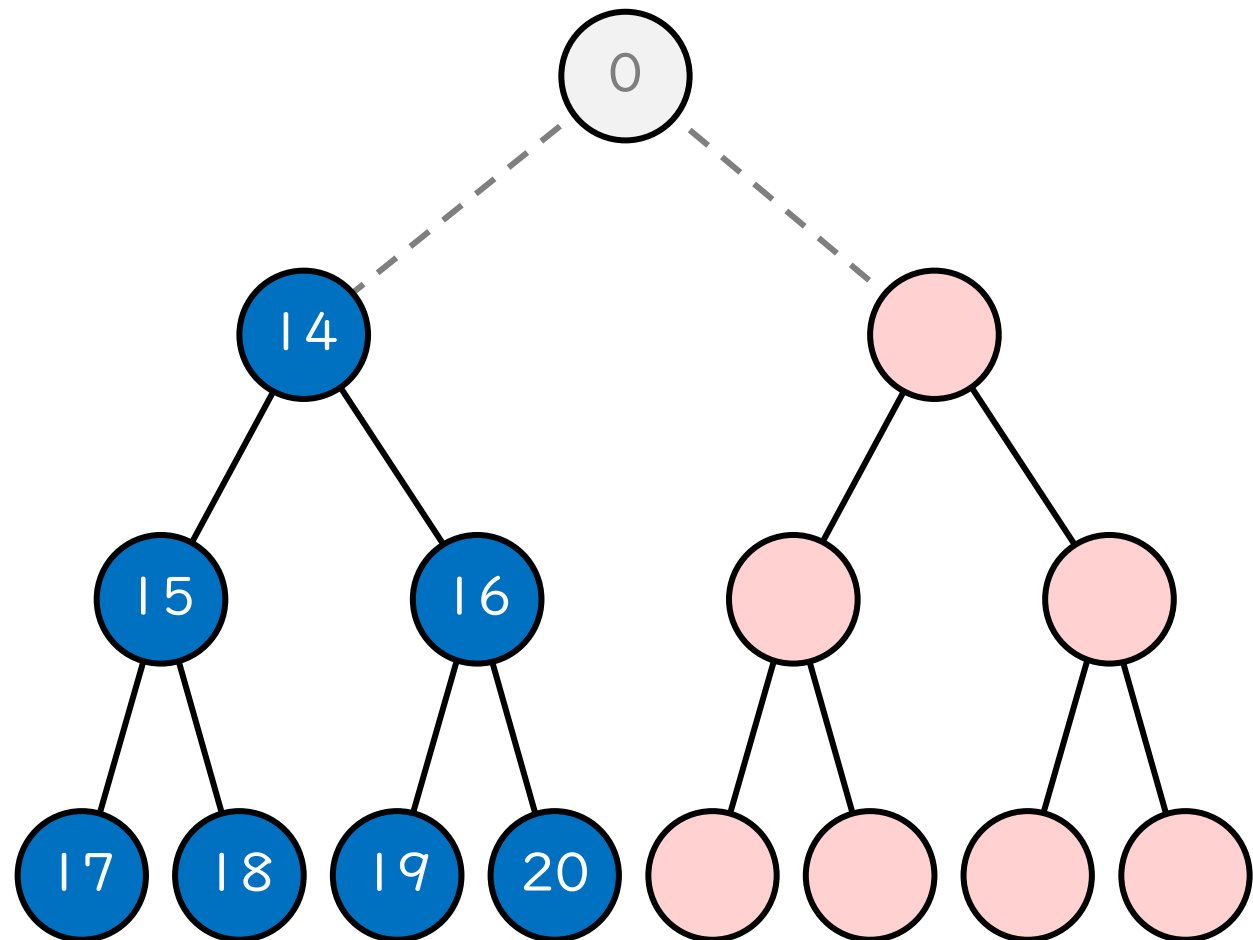


空港の ID の設定

$14 \times (\text{グループ番号})$
 $+ (\text{グループ内頂点番号})$

※グループ内頂点番号は適当に割り振る

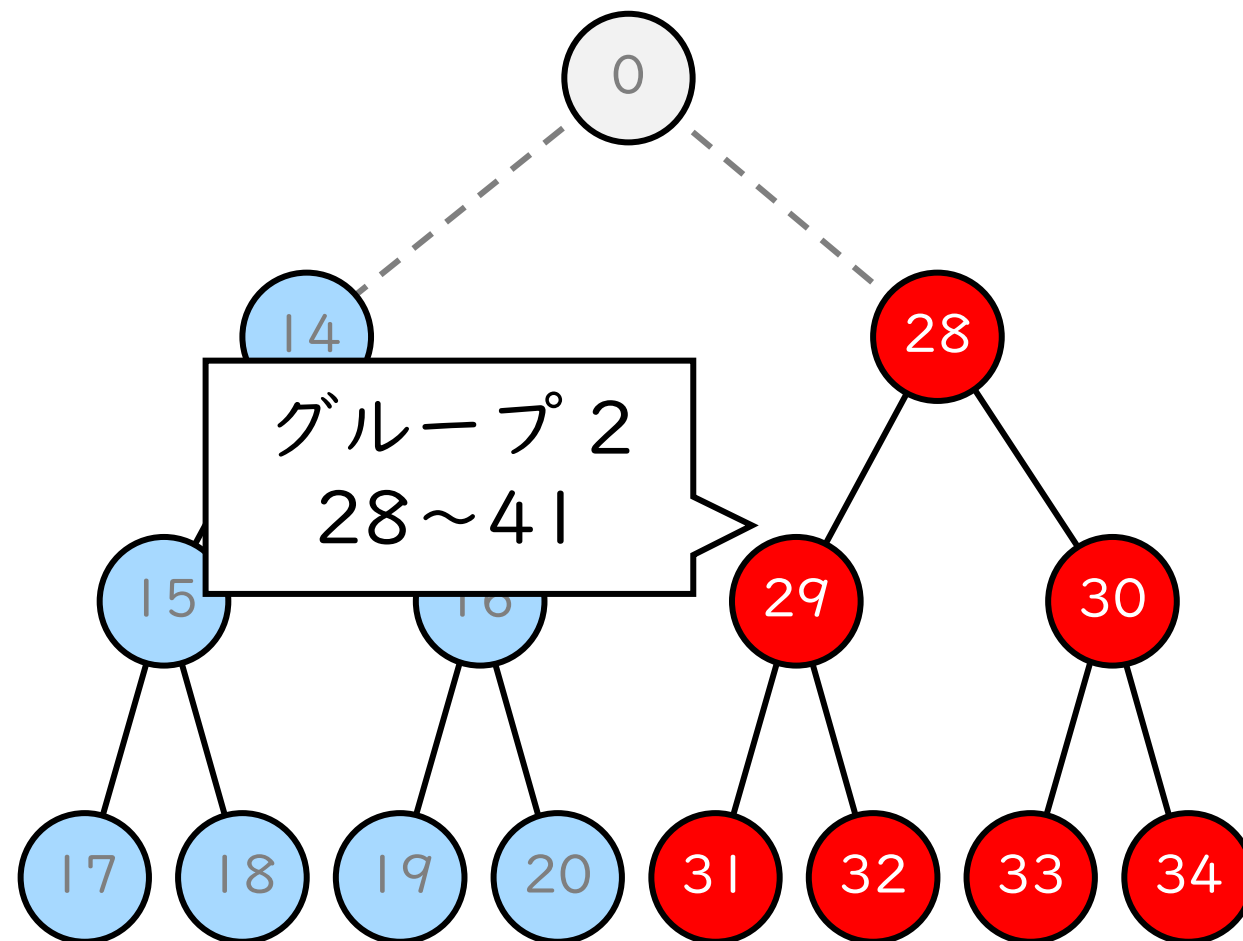
グループ 1
14 ~ 27



空港の ID の設定

$14 \times (\text{グループ番号})$
 $+ (\text{グループ内頂点番号})$

※グループ内頂点番号は適当に割り振る



ステップ 2

Ali → Benjamin の通信

Benjamin が送る情報

遊園地のグループ番号 G_X

温泉のグループ番号 G_Y

Benjamin が送る情報

遊園地のグループ番号 G_X 温泉のグループ番号 G_Y

グループの数は最大で

$$10000 \div 7 = 1429 \text{ 個}$$

識別すべきパターン数は

やばい、20 ビットで
送れない！

$$1429^2 > 2^{20}$$

3 59 点解法：ステップ 2

Benjamin が送る情報

遊園地のグループ番号 G_X

温泉のグループ番号 G_Y

グループの数は最大で

$$10000 \div 7 = 1429 \text{ 個}$$

識別すべきパターン数は

$$\cancel{1429^2} > \cancel{2^{20}}$$

$G_X \leq G_Y$ を仮定しても良い
(復習：32 点解法)

$$1429^2 \div 2 < 2^{20}$$

ステップ 3

Ali → Benjamin の通信

重要な性質

グループ間を行き来するために

絶対通るパスが存在

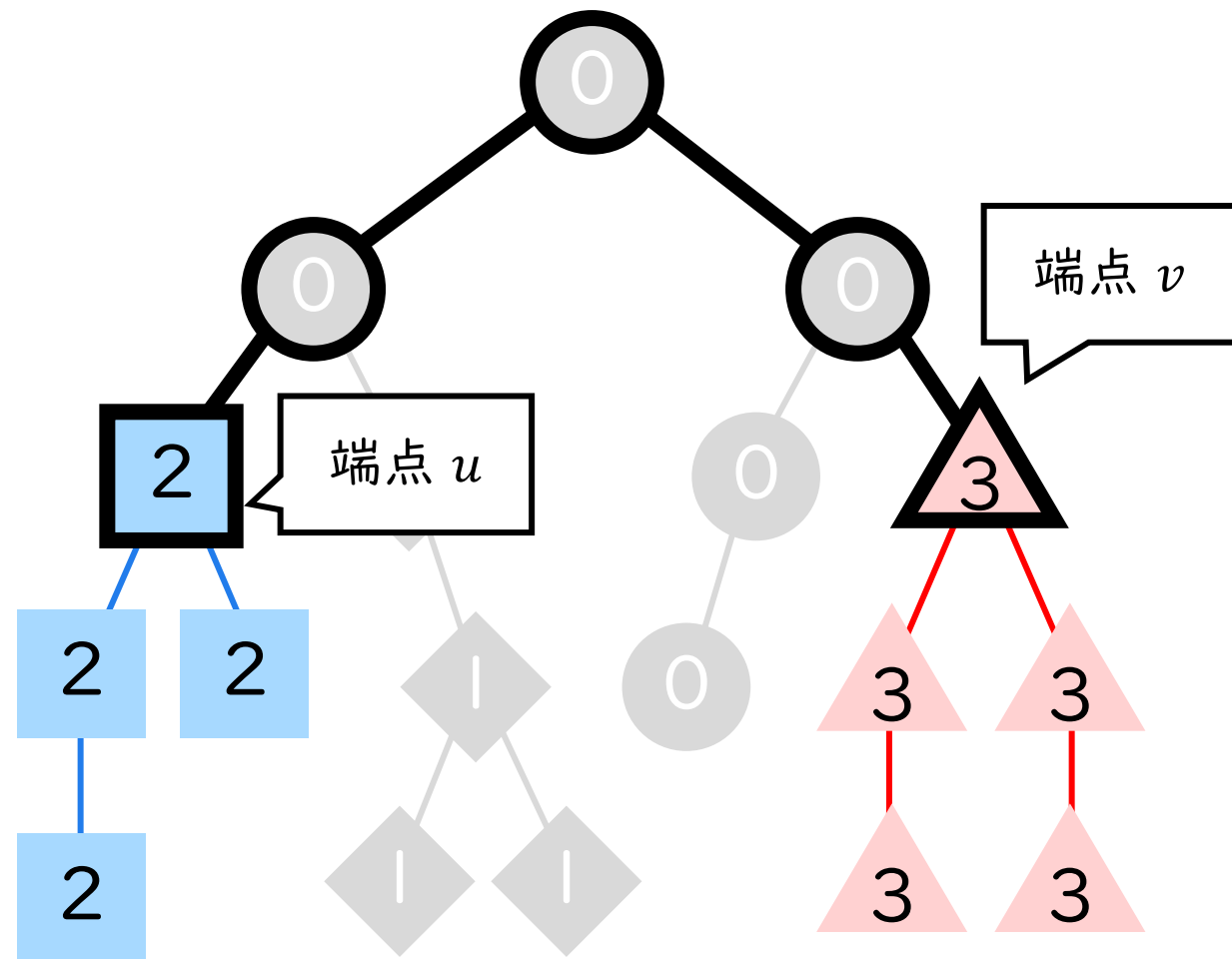
重要な性質

グループ間を行き来するために

絶対通るパスが存在

※遊園地（青色）側の端点を u

温泉（赤色）側の端点を v とする



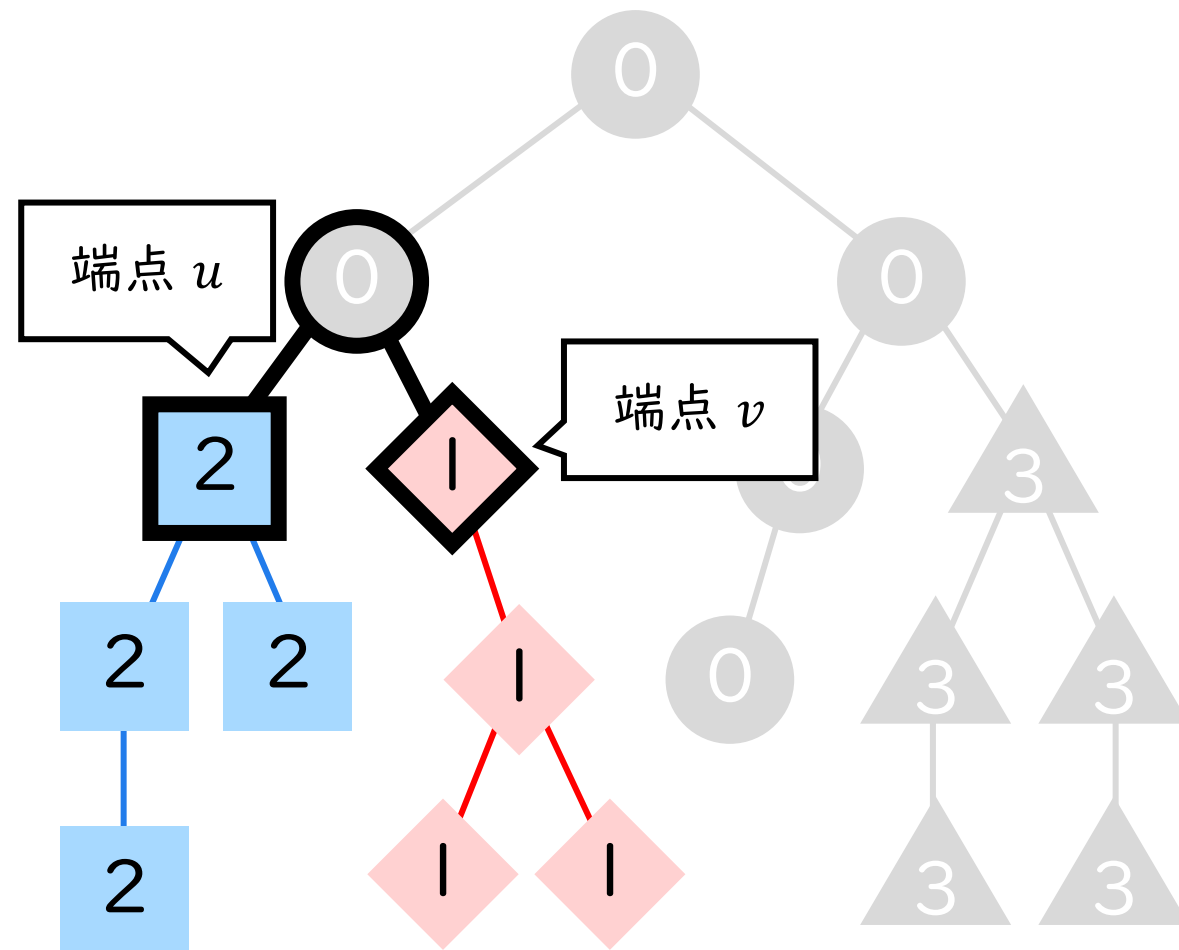
重要な性質

グループ間を行き来するために

絶対通るパスが存在

※遊園地（青色）側の端点を u

温泉（赤色）側の端点を v とする



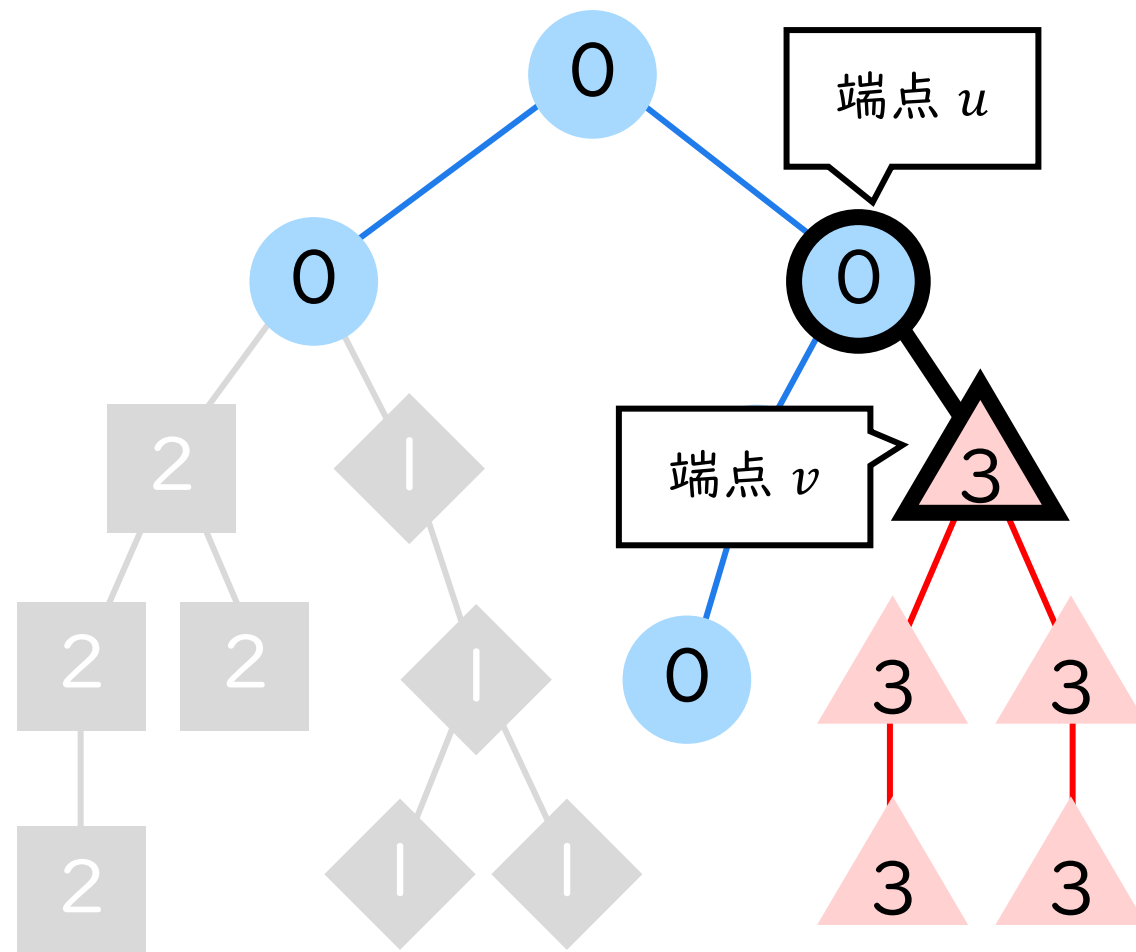
重要な性質

グループ間を行き来するために

絶対通るパスが存在

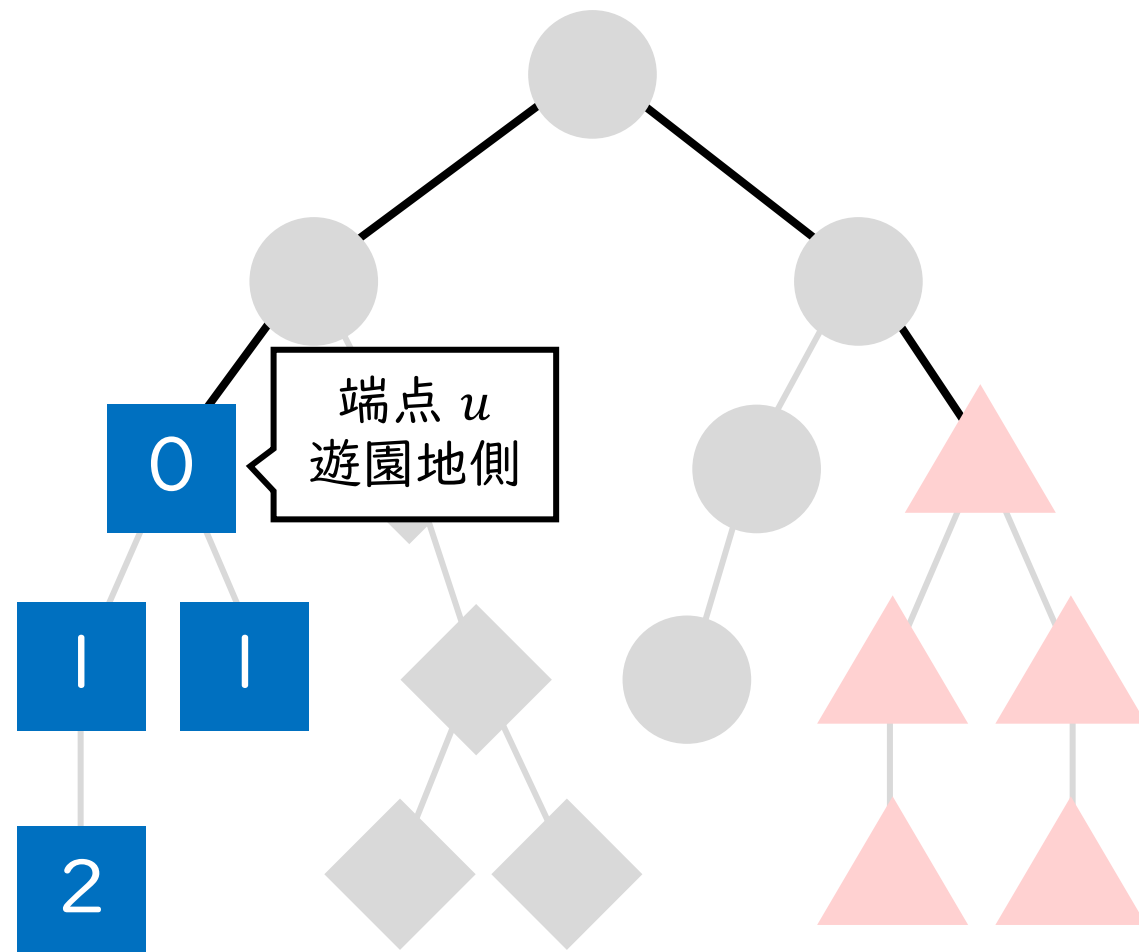
※遊園地（青色）側の端点を u

温泉（赤色）側の端点を v とする

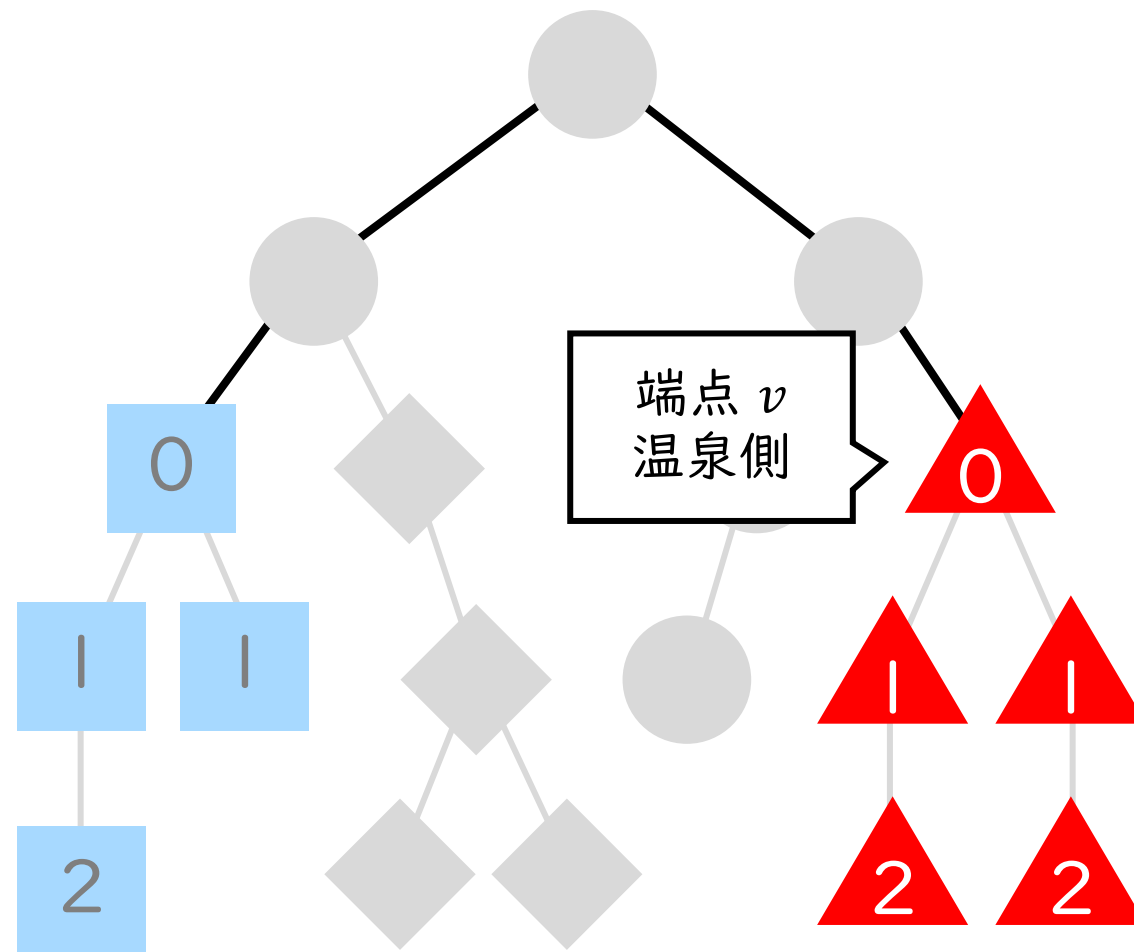


Q. Ali はどんな情報を
送れば良いのか？

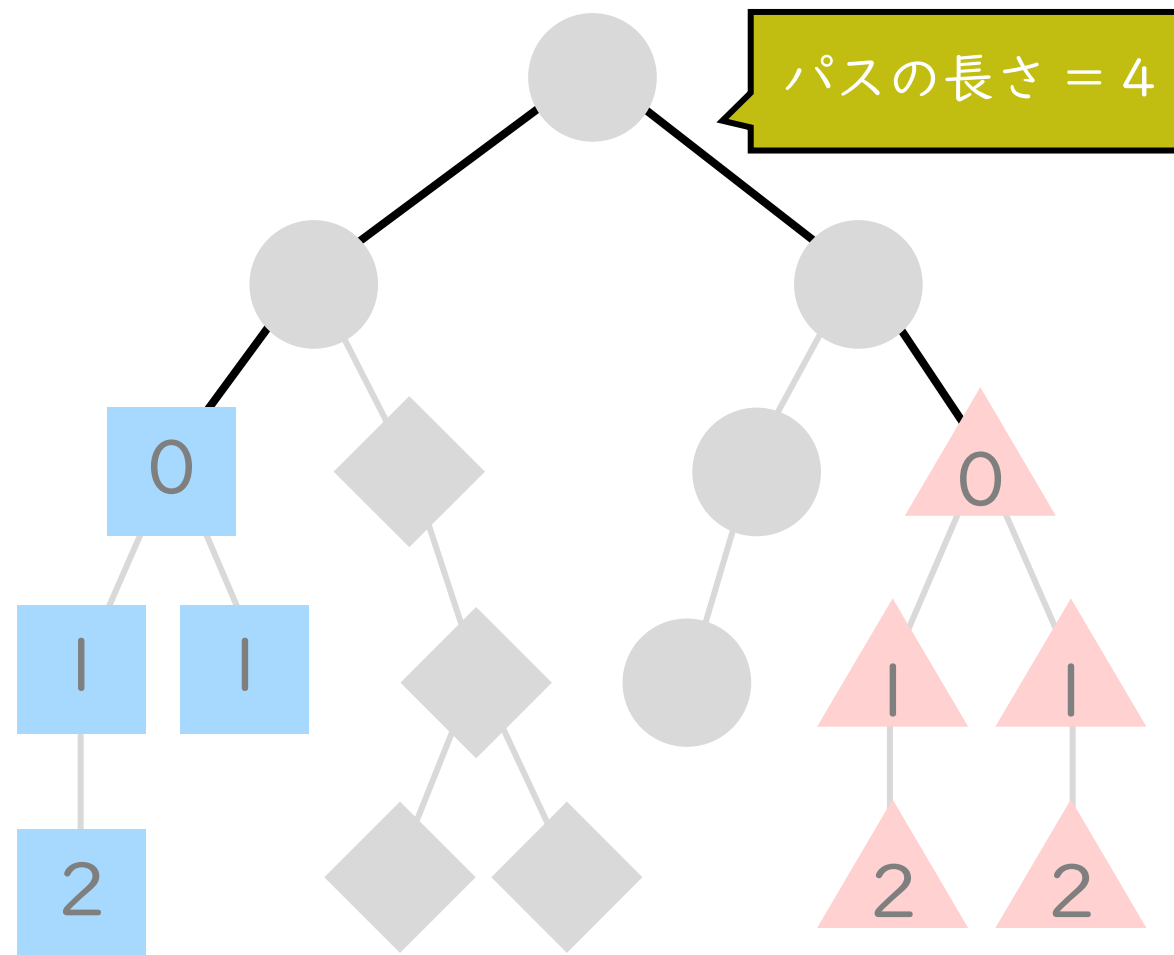
Ali が送る情報

1 端点 u から各頂点への距離2 端点 v から各頂点への距離3 $u - v$ 間の距離

Ali が送る情報

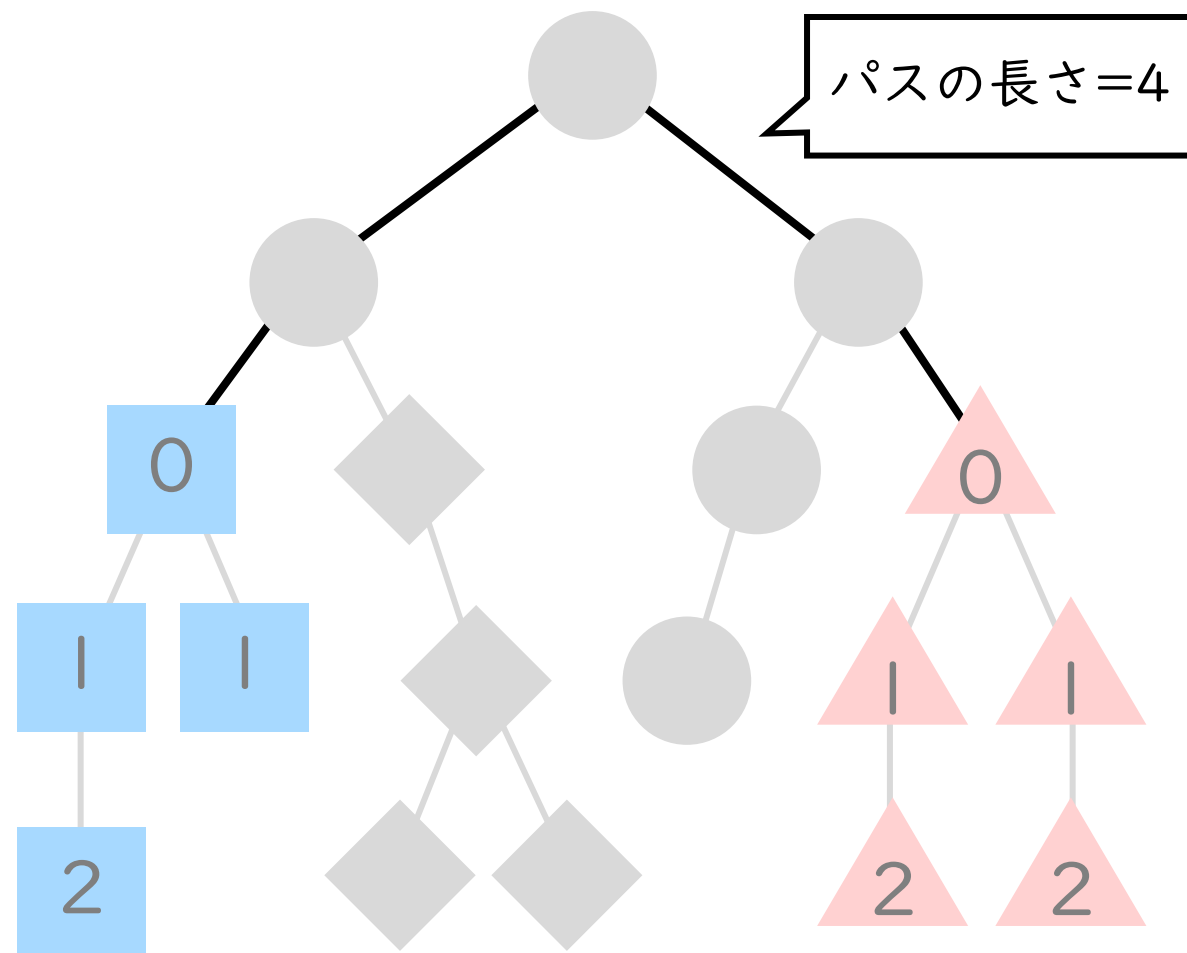
1 端点 u から各頂点への距離2 端点 v から各頂点への距離3 $u - v$ 間の距離

Ali が送る情報

1 端点 u から各頂点への距離2 端点 v から各頂点への距離3 $u - v$ 間の距離

本当にこの情報だけで
最短距離が求められるのか？

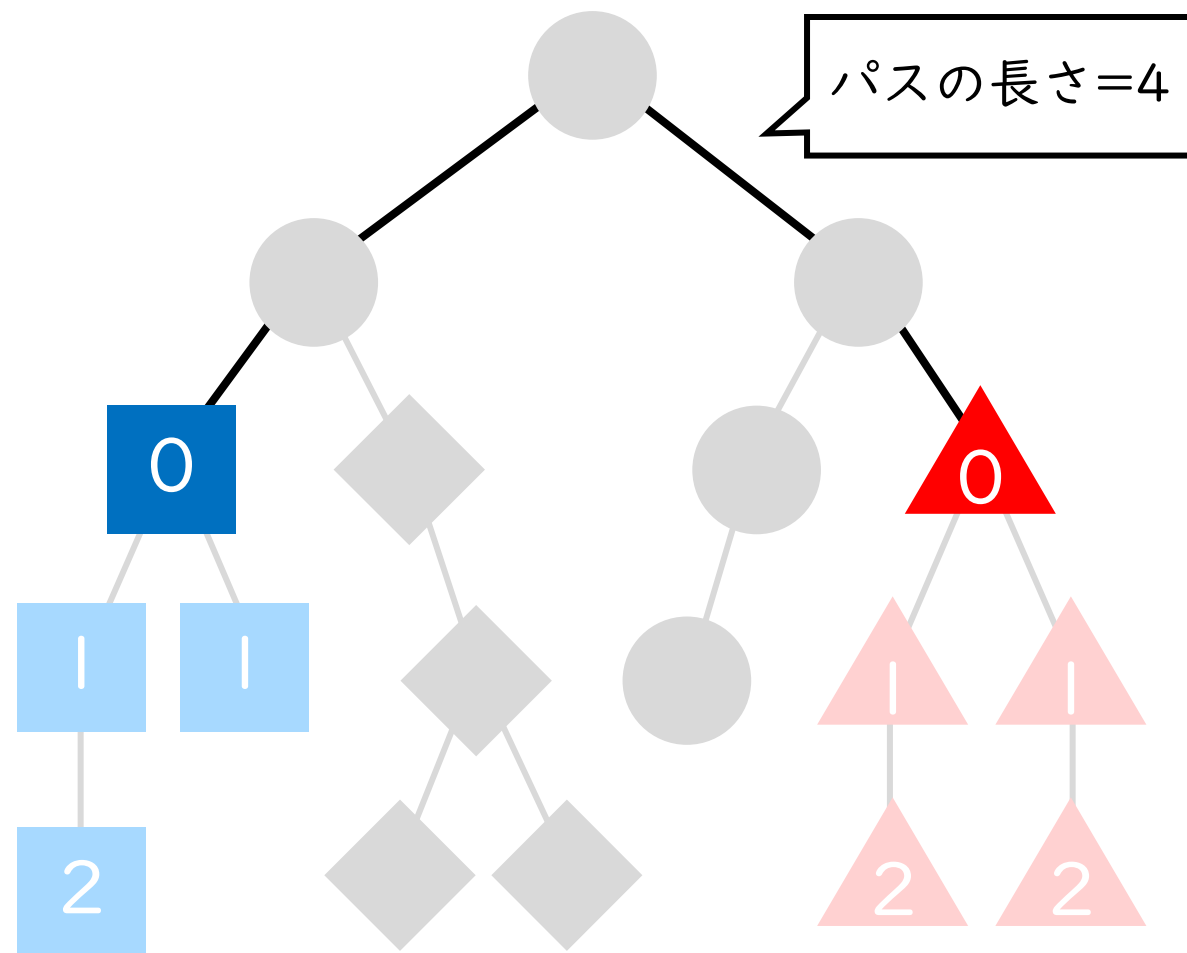
実は全部足すだけで
最短距離が求められる



実は全部足すだけで
最短距離が求められる

最短距離

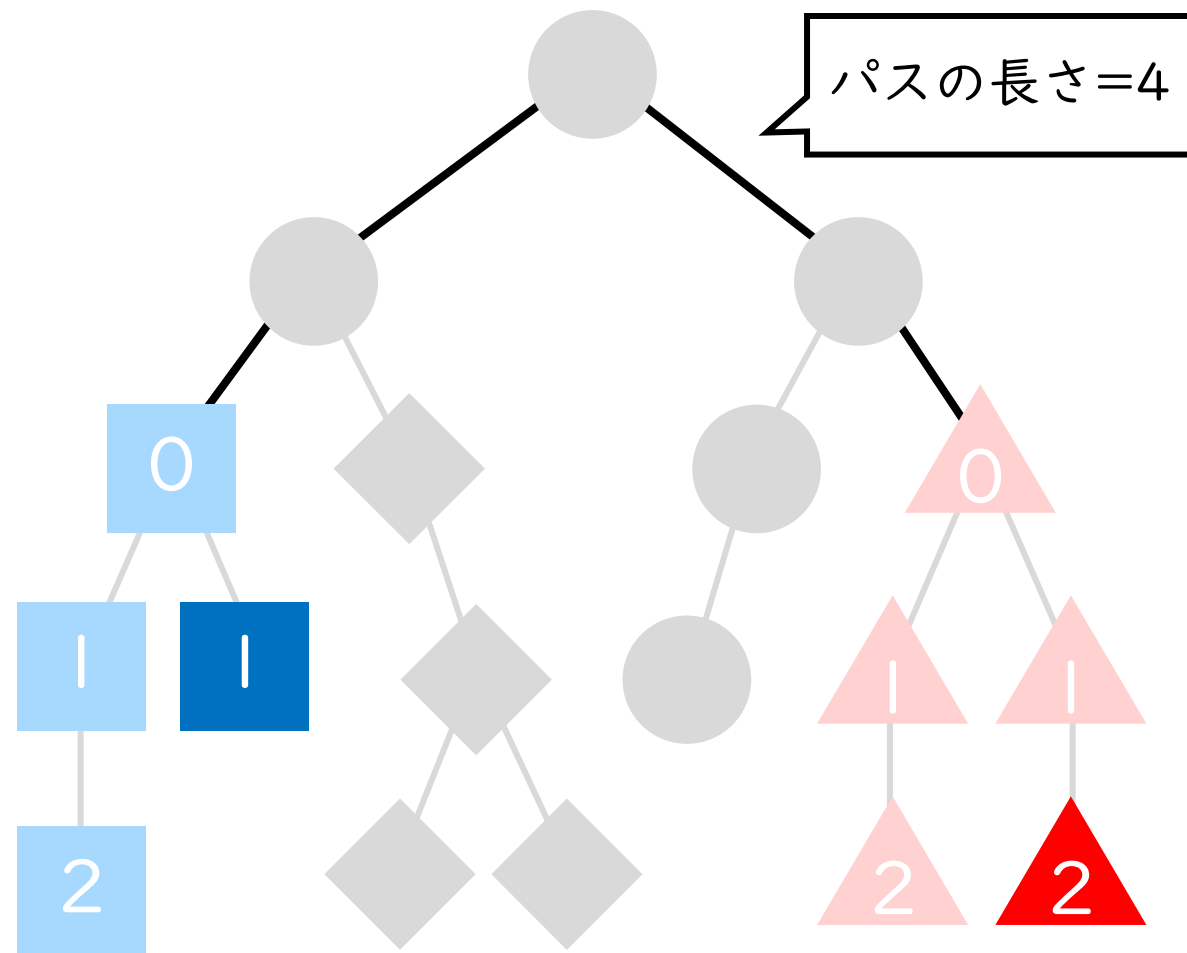
$$4 + 0 + 0 = 4$$



実は全部足すだけで
最短距離が求められる

最短距離

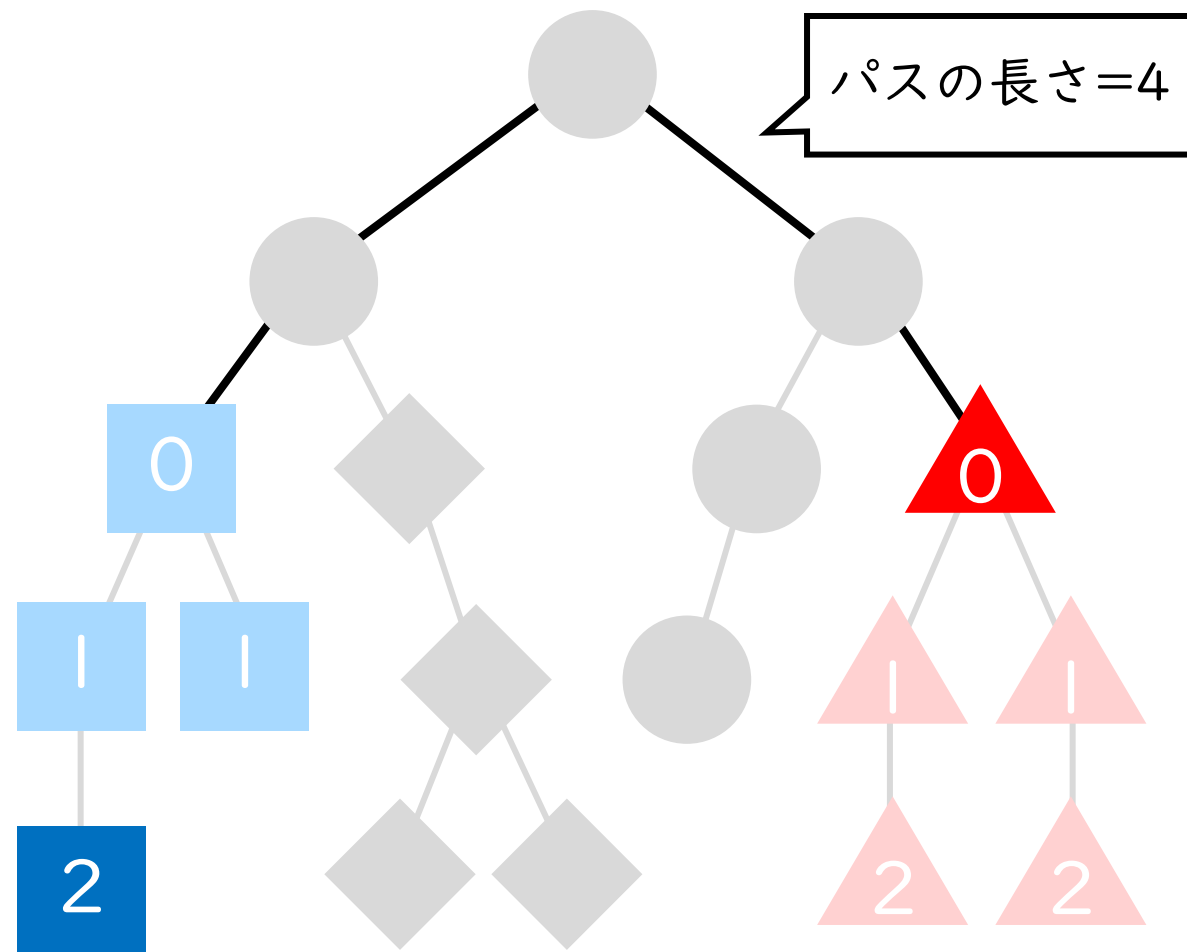
$$4 + 1 + 2 = 7$$



実は全部足すだけで
最短距離が求められる

最短距離

$$4 + 2 + 0 = 6$$



最後に、Alice は
何ビット送る必要があるか？

3 59 点解法：ステップ 3

必要なビット数は？

1. 端点 u からの距離

$$\rightarrow 13 \text{ 個} \times 4 \text{ ビット} = 52 \text{ ビット}$$

一つのグループの
最大頂点数は 13

最大距離は 13
 $2^4=16$ で十分

3 59 点解法：ステップ 3

必要なビット数は？

1. 端点 u からの距離 $\rightarrow 13$ 個 $\times 4$ ビット = 52 ビット

2. 端点 v からの距離 $\rightarrow 13$ 個 $\times 4$ ビット = 52 ビット

一つのグループの
最大頂点数は 13

最大距離は 13
 $2^4=16$ で十分

3 59 点解法：ステップ 3

131
—
222

必要なビット数は？

1. 端点 u からの距離

$$\rightarrow 13 \text{ 個} \times 4 \text{ ビット} = 52 \text{ ビット}$$

2. 端点 v からの距離

$$\rightarrow 13 \text{ 個} \times 4 \text{ ビット} = 52 \text{ ビット}$$

3. $u - v$ 間の距離

$$\rightarrow 14 \text{ ビット}$$

合計ビット数
118

3 59点解法：ステップ3

必要なビット数は？

1. 端点 u からの距離 $\rightarrow 13$ 個 $\times 4$ ビット = 52ビット

ここまでの点数：**59**点

2. 端点 v からの距離 $\rightarrow 13$ 個 $\times 4$ ビット = 52ビット

3. $u-v$ 間の距離 $\rightarrow 14$ ビット

合計ビット数
118

Stage 4

40 ビットまでの解法

Q. もっと効率的に
各グループの情報を送れないか？

11 点解法

Benjamin が頂点番号 X を送り

Ali が番号 X から各頂点への距離を送る



合計ビット数
14万

11 点解法

Benjamin が頂点番号 X を送り
Ali が番号 X から各頂点への距離を送る



合計ビット数
14万

15 点解法

Ali が木を $2(N - 1)$ ビットの文字列として送る
Benjamin は何もしない



合計ビット数
2万

11 点解法

グループ内の距離を送るより

Benjamin が頂点番号 X を送り木を $2(N-1)$ ビットで送った方が得なのでは？

合計ビット数

14万

15 点解法

検証してみよう

Ali が木を $2N$ ビットで

Benjamin は何もしない

合計ビット数

2万

Ali が必ず送るべき情報は以下の 4 つ

1

遊園地グループの木の情報

2

温泉グループの木の情報

3

端点 $u-v$ 間の距離

4

端点 u, v のグループ内番号

Ali が必ず送るべき情報は以下の 4 つ

1

遊園地グループの木の情報

2

温泉グループの木の情報

3

端点 $u-v$ 間の距離

4

端点 u, v のグループ内番号

木の情報を送っただけでは

最短距離が求められないことに注意

Alice は何ビット送る必要があるか？

Ali が必ず送るべき情報は以下の 4 つ

最大頂点数は $N = 13$
ビット数は $2(N - 1) = 24$

1

遊園地グループの木の情報

$$\rightarrow 2 \times 12 = 24 \text{ ビット}$$

2

温泉グループの木の情報

$$\rightarrow 2 \times 12 = 24 \text{ ビット}$$

3

端点 $u - v$ 間の距離

$$\rightarrow 14 \text{ ビット}$$

4

端点 u, v のグループ内番号

$$\rightarrow 2 \times 4 = 8 \text{ ビット}$$

合計ビット数

70



Ali が必ず送るべき情報は以下の 4 つ

最大頂点数は $N = 13$
ビット数は $2(N - 1) = 24$

1 遊園地グループの木の情報

→ $2 \times 12 = 24$ ビット

2 ここまでの点数:

67 点

3 端点 $u-v$ 間の距離

→ 14 ビット

4 端点 u, v のグループ内番号

→ $2 \times 4 = 8$ ビット

合計ビット数

70

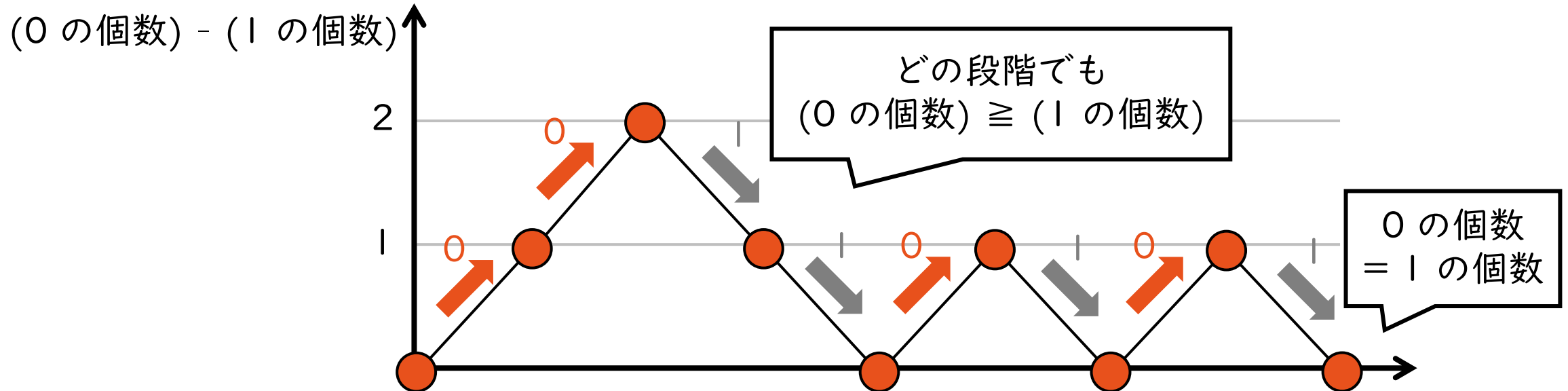
長さ $2(N - 1)$ の 01 文字列すべてが
木を表すとは限らない

× “11111111”

× “00111100”

重要な考察

木を表す文字列は **カッコ列** と同じ条件を満たす



パターン数はどれくらいか？

長さ 24 の文字列 : 2^{24} 通り ▶ 24 ビット

パターン数はどれくらいか？

※カッコ列の数はカタラン数と関連します。
興味のある方は調べてみてください。

~~長さ 24 の文字列 : 224 通り ▶ 24 ビット~~

長さ 24 以下のカッコ列 : 29万 通り ▶ 19 ビット

合計ビット数

60



パターン数はどれくらいか？

※カッコ列の数はカタラン数と関連します。
興味のある方は調べてみてください。

ここまでの点数：**72** 点

長さ 24 以下のカッコ列：29万 通り

19 ビット

合計ビット数

60

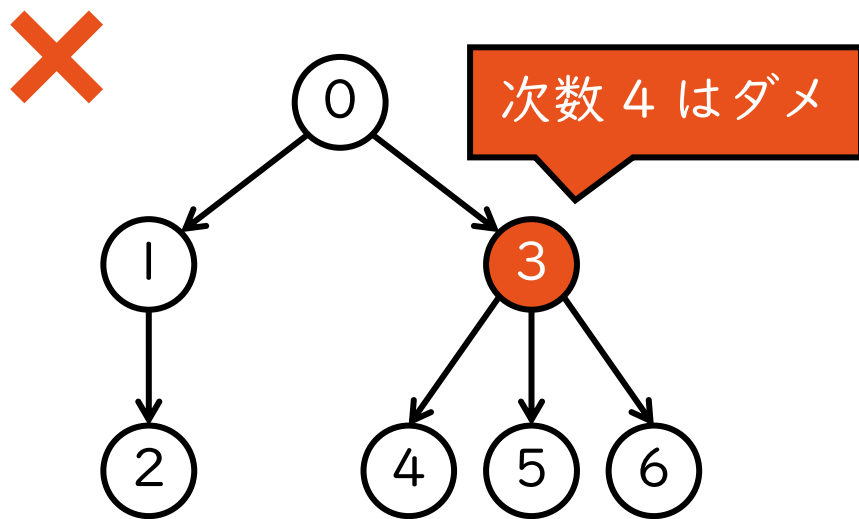
実は木のパターン数を
さらに減らすことができます

ポイント 1

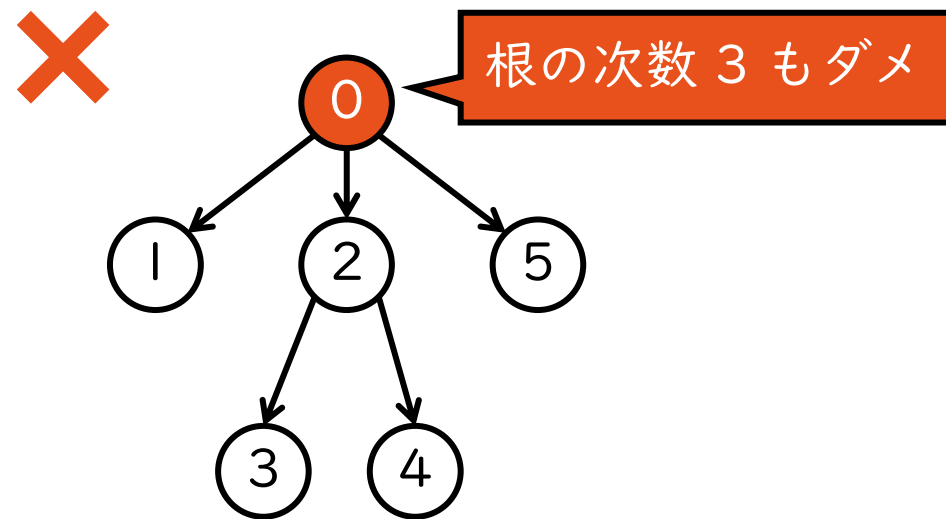
根付き木の次数は 2 以下

ポイント 1

根付き木の出次数は 2 以下



“001100101011”



“0100101101”

ポイント 1

根付き木の次数は 2 以下

ポイント 2

根以外の部分木のサイズは 6 以下

※サイズ 7 だと貪欲法では切られてしまう！

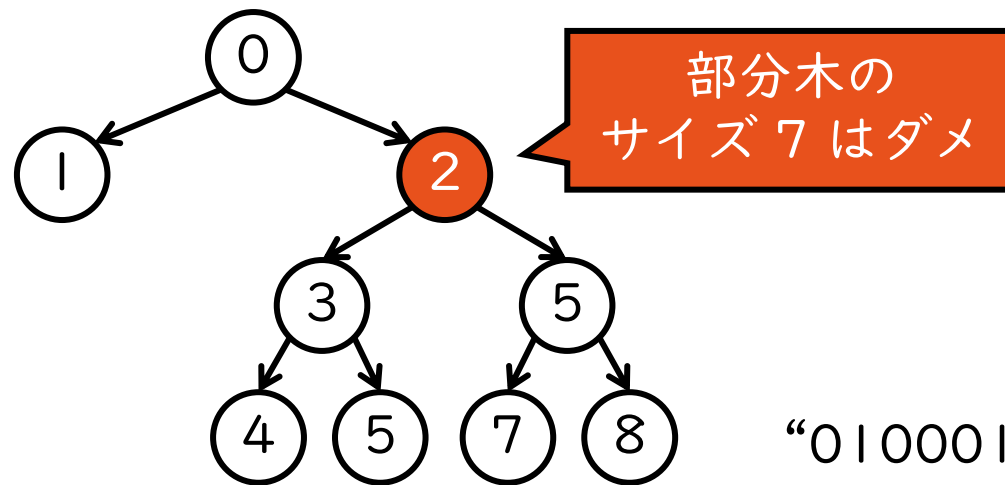
ポイント 1

根付き木の出次数は 2 以下

ポイント 2

根以外の部分木のサイズは 6 以下

※サイズ 7 だと貪欲法では切られてしまう！



“0100010110010111”

パターン数はどれくらいか？

長さ 24 以下のカッコ列： **29万** 通り ▶ **19** ビット



パターン数はどれくらいか？

合計ビット数

40

ここまでの点数：**84** 点

条件を満たす木の個数：325 通り ▶ 9 ビット

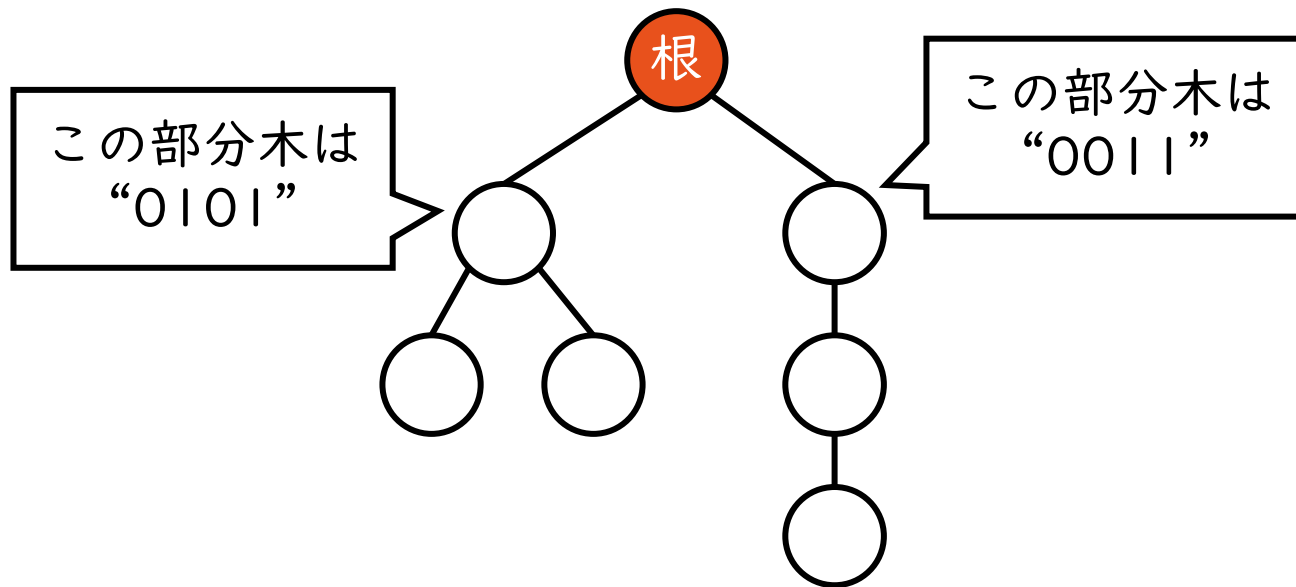
※同型な木を同じとみなさない場合は 1483 通り (11 ビット、80 点)

84 点を獲得するためには、同型な木を「同じもの」だとしてビットを送らなければならない（そうしなければ 80 点になる）

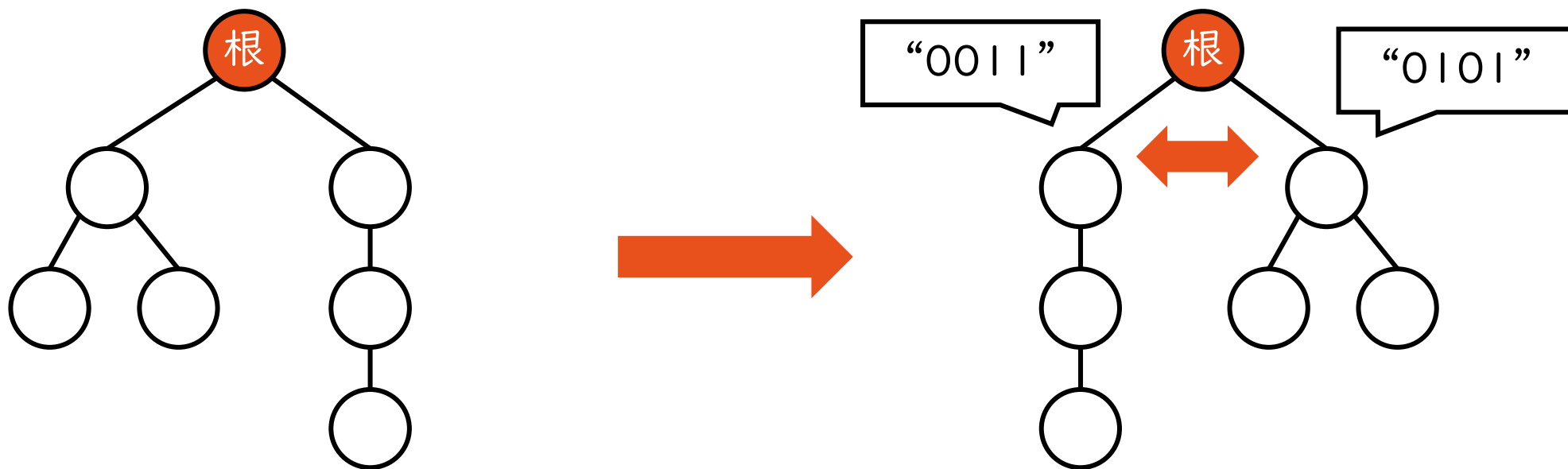


2つの木が同型かどうかを判定するためには、部分木を表す文字列の辞書順に辺を付け替えれば良い

2つの木が同型かどうかを判定するためには、**部分木を表す文字列の辞書順**に辺を付け替えれば良い



2つの木が同型かどうかを判定するためには、**部分木を表す文字列の辞書順**に辺を付け替えれば良い



Stage 5

29 ビットまでの解法

現時点で使っているビット数は、次の通り

1	遊園地グループの木の情報	9ビット
2	温泉グループの木の情報	9ビット
3	端点 $u-v$ 間の距離	14ビット
4	端点 u, v のグループ内番号	8ビット

現時点で使っているビット数は、次の通り

1	遊園地グループの木の情報	9ビット
2	温泉グループの木の情報	9ビット
3	端点 $u-v$ 間の距離	14ビット
4	端点 u, v のグループ内番号	8ビット → これを減らせないか？

重要な性質

端点 u, v のうち少なくとも木全体の根から遠い方は…

グループ内の根

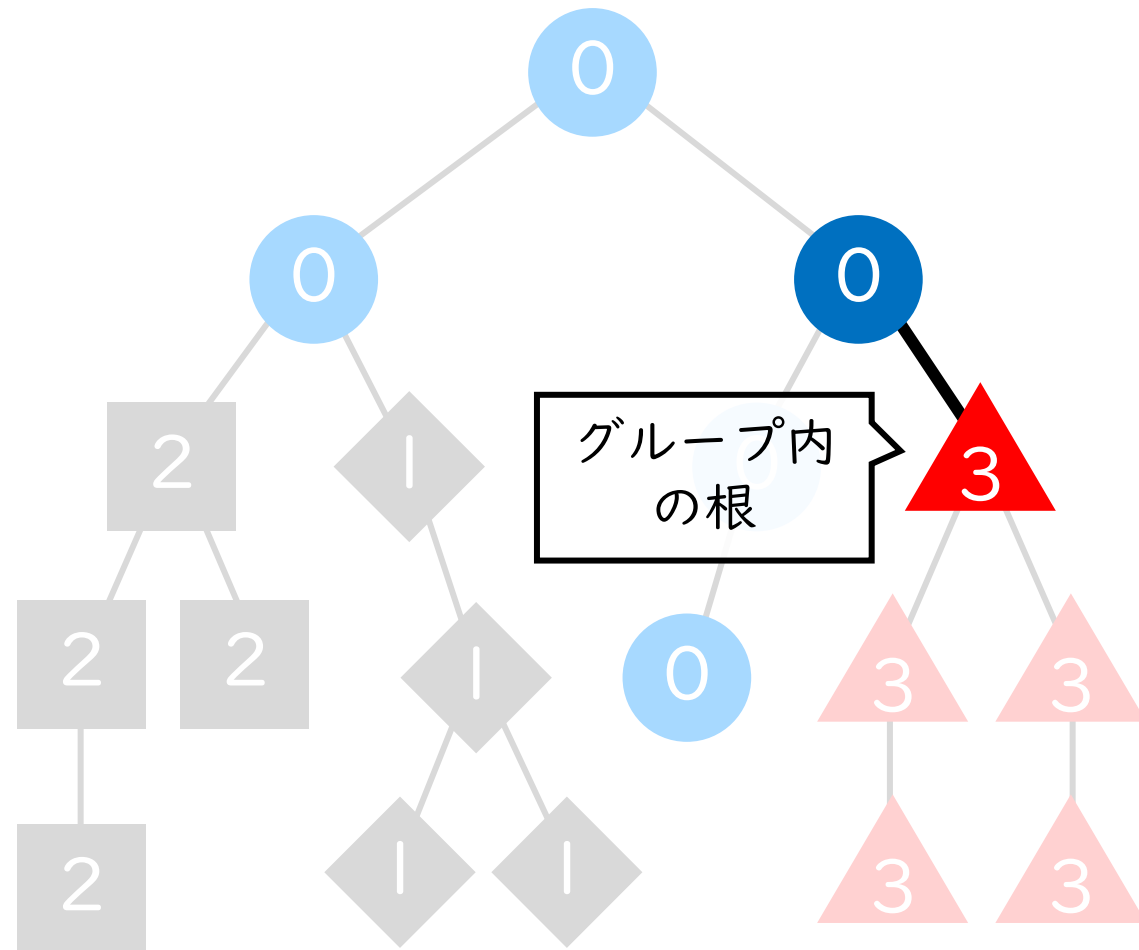
になっている

重要な性質

端点 u, v のうち少なくとも木全体の根から遠い方は...

グループ内の根

になっている

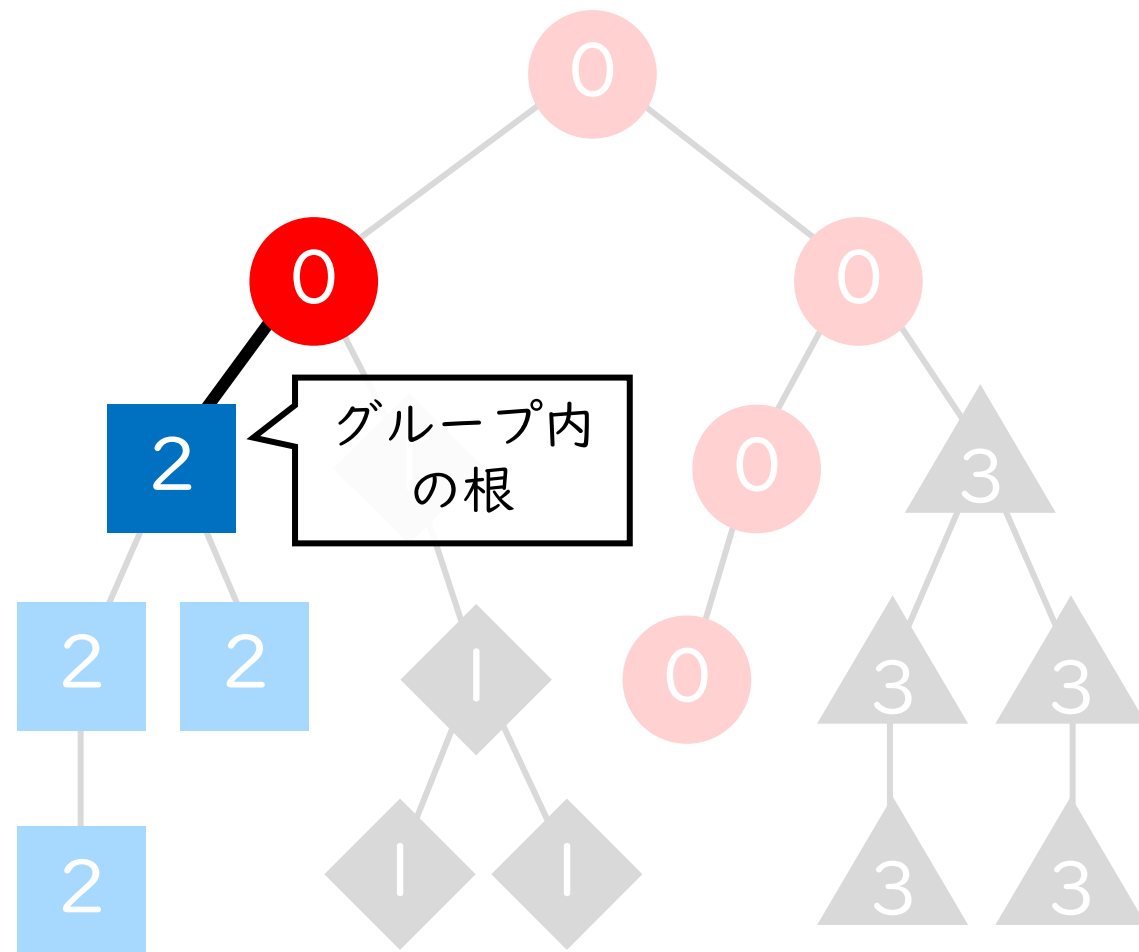


重要な性質

端点 u, v のうち少なくとも木全体の根から遠い方は...

グループ内の根

になっている

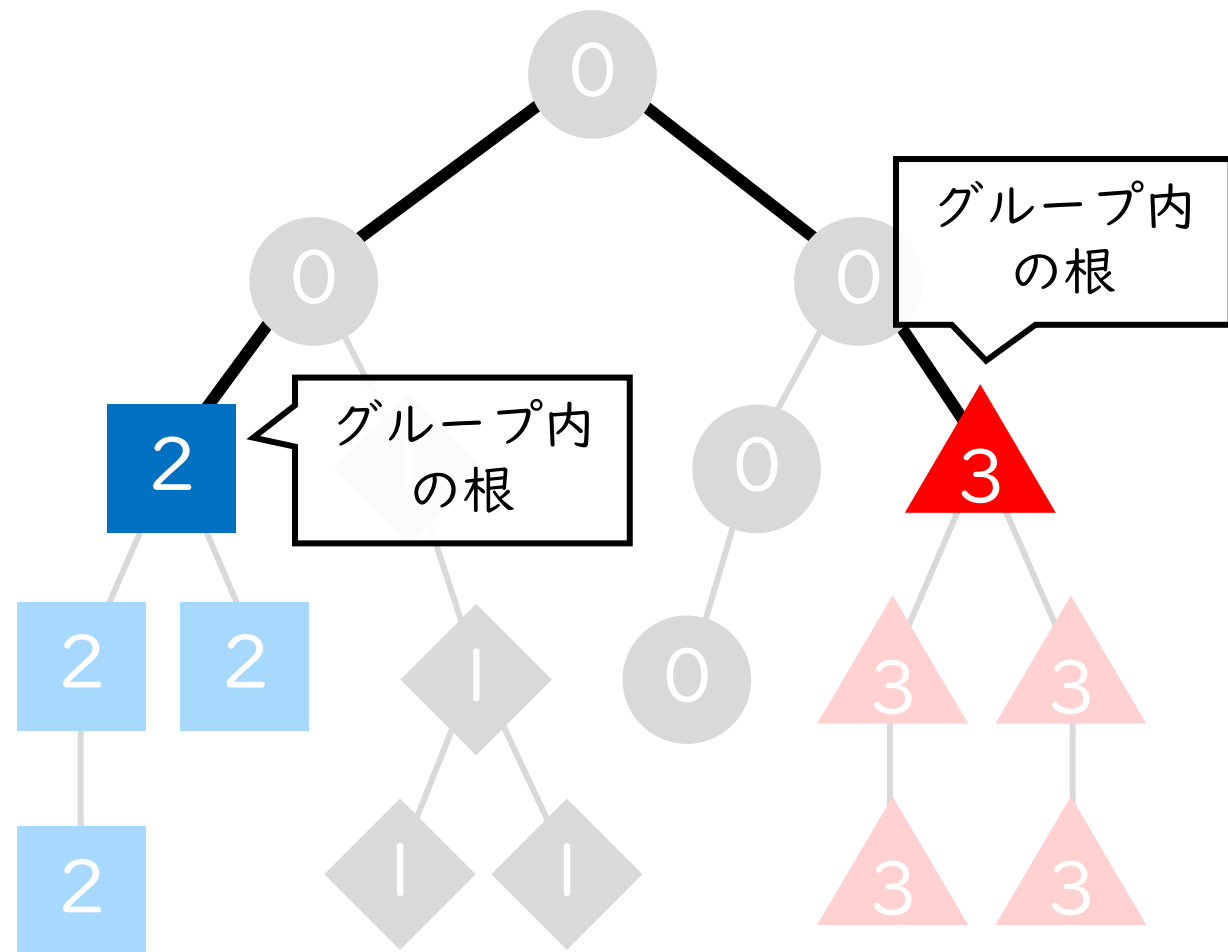


重要な性質

端点 u, v のうち少なくとも木全体の根から遠い方は…

グループ内の根

になっている



この性質を使うと…

端点 u, v のグループ内頂点番号のうち
グループ内の根である方は送らなくて良い

この性質を使うと…

端点 u, v のグループ内頂点番号のうち
グループ内の根である方は送らなくて良い



4 ビット減る！

合計ビット数
36

※ただし、 u と v どちらが「グループ内の根」であるかを判別するために
木全体の根に近い方のグループ番号を小さくするなどの工夫が必要



この性質を使うと…

端点 u, v のグループ内頂点番号のうち
グループ内の根である方は送らなくて良い

ここまでの点数：**88** 点

4 ビット減る！

合計ビット数

36

※ただし、 u と v どちらが「グループ内の根」であるかを判別するために
木全体の根に近い方のグループ番号を小さくするなどの工夫が必要

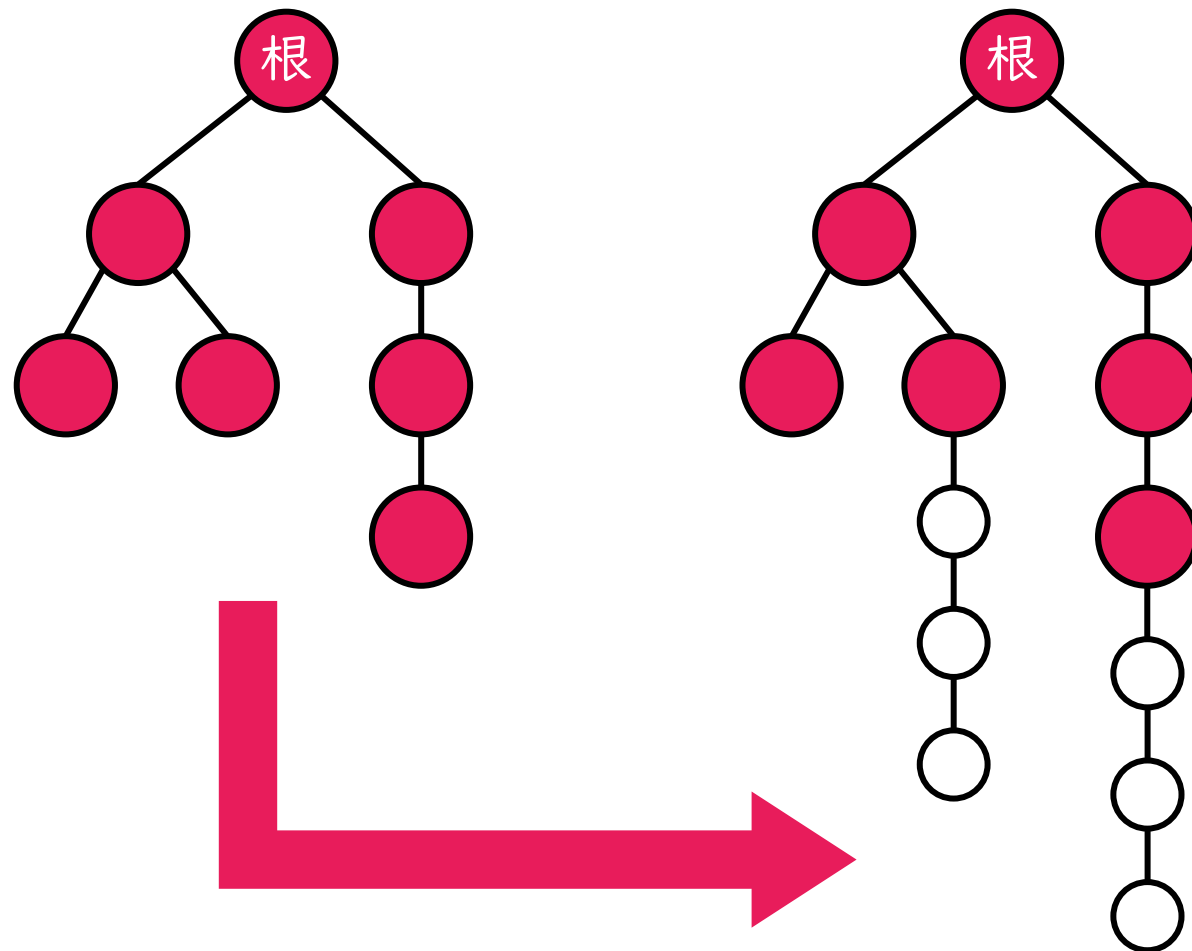
実は 1~13 頂点の木
すべてを考える必要はない

重要な性質

各グループの木に頂点を適当に追加することで、

- 次数 ≤ 3
- 部分木サイズ ≤ 6

を満たす 13 頂点の木が生成できる





パターン数はどれくらいか？

13 頂点以下の木： 325 通り ▶ 9 ビット

パターン数はどれくらいか？

~~13 頂点以下の木 : 325 通り ▶ 9 ビット~~

13 頂点の木 : 66 通り ▶ 7 ビット

合計ビット数
32



パターン数はどれくらいか？

ここまでの点数：**92** 点

13 頂点の木

： 66 通り

▶ 7 ビット

合計ビット数

32

これまでは、以下の 4 つの情報を独立に送っていた

情報	パターン数
1 つ目の木の情報	66 通り
2 つ目の木の情報	66 通り
接点のグループ内頂点番号	14 通り
接点 u, v 間の距離	10000 通り

➡ 7 ビット

➡ 7 ビット

➡ 4 ビット

➡ 14 ビット

これらをまとめて送ると 30 ビットまで減る

情報	パターン数
1 つ目の木の情報	66 通り
2 つ目の木の情報	66 通り
接点のグループ内頂点番号	14 通り
接点 u, v 間の距離	10000 通り

$$66 \times 66 \times 14 \times 10000 \\ = 609840000 \text{ 通り}$$

30 ビットで十分

これらをまとめて送ると 30 ビットまで減る

情報	パターン数
1つ目の木の情報	66 通り
2つ目の木の情報	66 通り
接点のグループ内頂点番号	14 通り
接点 u, v 間の距離	10000 通り

ここまでの点数 :

94 点

$$66 \times 66 \times 14 \times 10000 = 609840000 \text{ 通り}$$

30 ビットで十分

これまでは Ali が 30 ビットの情報しか送らなかった



Ali が 1 ビットから **29** ビットまで全部使った場合

ほぼ 2^{30} 通りを識別できる※

合計ビット数
29

これまでは A_i が 30 ビットの情報しか送らなかった

ここまでの点数：**95**点

A_i が 1 ビットから 29 ビットまで全部使った場合

ほぼ 2^{30} 通りを識別できる※

合計ビット数

29

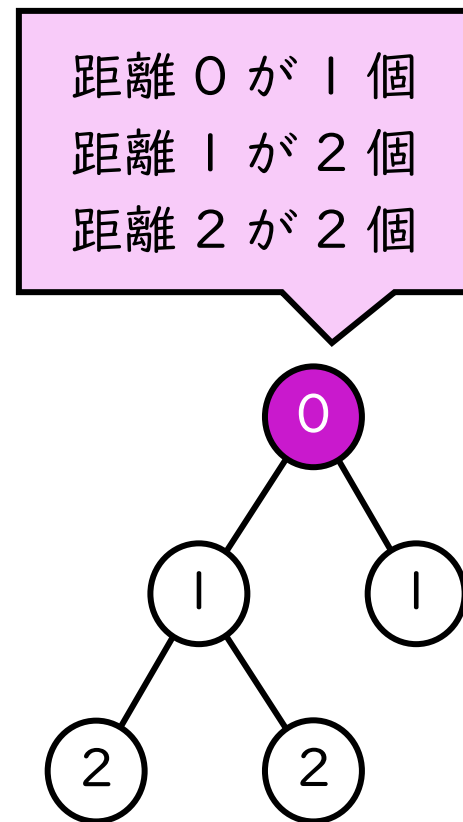
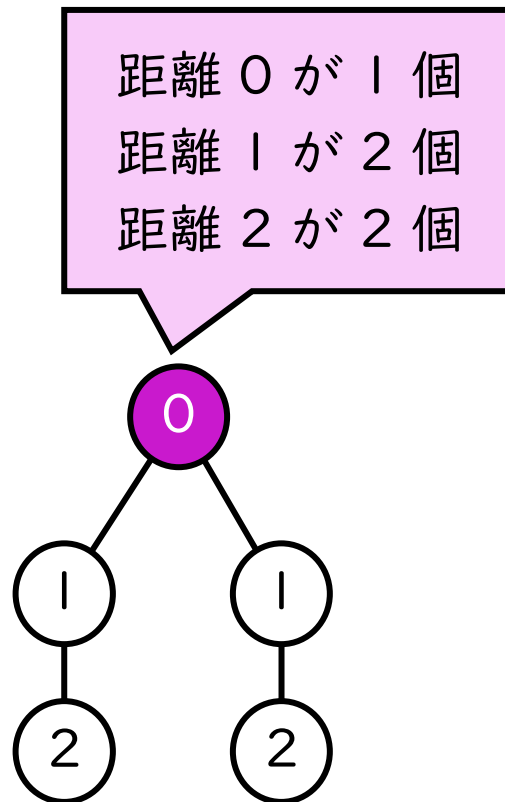
※ $2^1 + 2^2 + 2^3 + \dots + 2^{29} = 2^{30} - 2$

Extra Stage

満点 (24 ビット) へ

重要な性質

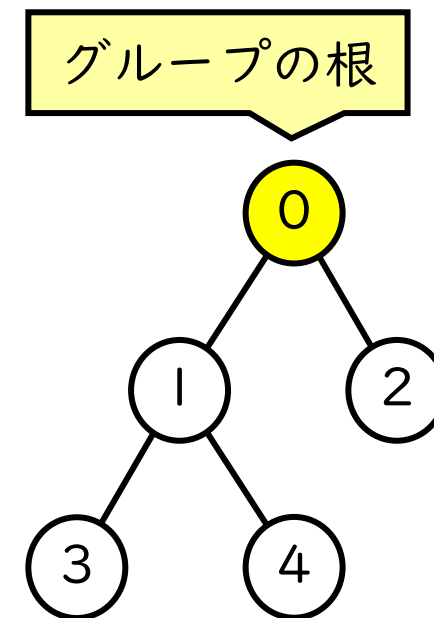
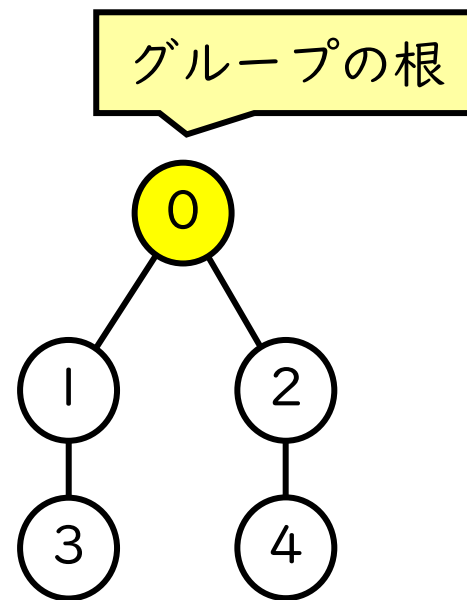
- 実は、木の代わりに距離分布を送った方が得
- たとえば以下の 2 つの木を「同じ」とみなせる



具体的にどうやって実現するか？

手順 I

各グループについて
グループ内番号を BFS 順に設定



手順 1

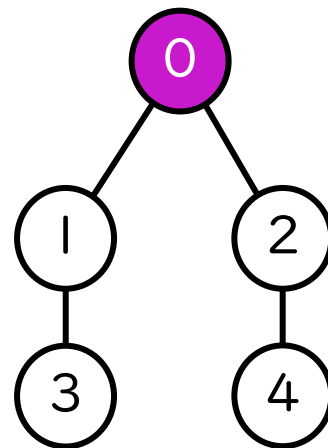
各グループについて
グループ内番号を BFS 順に設定

手順 2

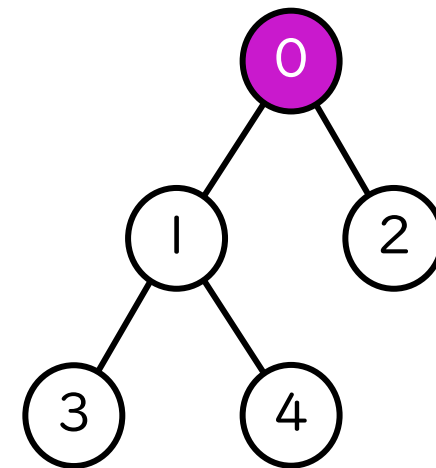
以下の組を送る

- 端点のグループ内番号
- 端点から ID=0 までの距離
:
- 端点から ID=12 までの距離

端点 0 からの距離
ID=0 から順に
(0, 1, 1, 2, 2)



端点 0 からの距離
ID=0 から順に
(0, 1, 1, 2, 2)



手順 1

各グループについて
グループ内番号を BFS 順に設定

端点 0 からの距離
ID=0 から順に
(0, 1, 1, 2, 2)

端点 0 からの距離
ID=0 から順に
(0, 1, 1, 2, 2)

異なる木でも同じ情報を

手順 2

以下の組を送る

- 端点のグループ内番号
- 端点から ID=0 までの距離
- :
- 端点から ID=12 までの距離

送っていて得である！



パターン数はどれくらいか？

端点が根の場合

木を送る 距離分布を送る

66通り → 38通り

パターン数はどれくらいか？

端点が根の場合

端点が根とは限らない場合

木を送る 距離分布を送る

66通り → 38通り

858通り → 676通り※

[66×13通り]

※38×13通りまでは行かないが、66×13通りよりは少ない



パターン数はどれくらいか？

端点が根の場合

27

ビット

97

点

端点が根とは限らない場合

木を送る

距離分布を送る

66通り \rightarrow 38通り

木を送る

距離分布を送る

858通り \rightarrow 676通り※

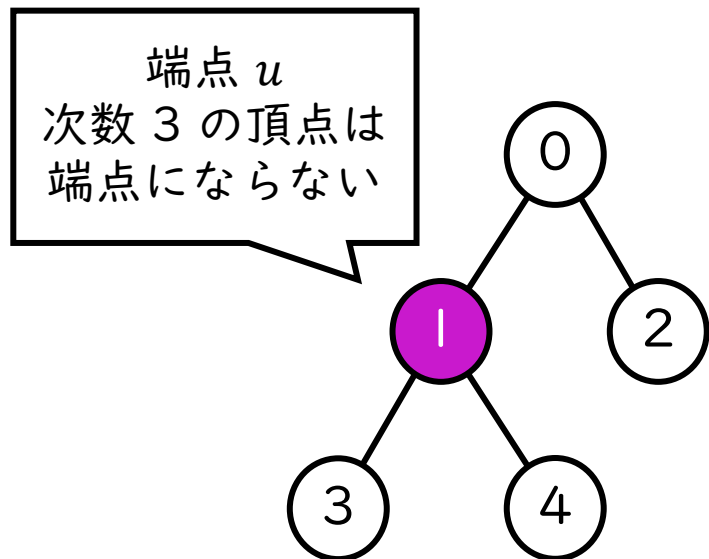
[66×13通り]

※38×13 通りまでは行かないが、66×13 通りよりは少ない

重要な考察

端点と木の組合せとして

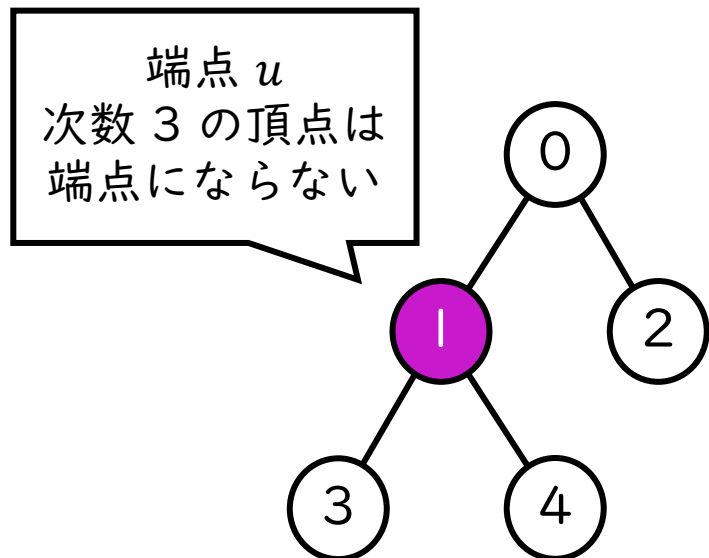
以下のようなものは絶対にあり得ない



重要な考察

端点と木の組合せとして

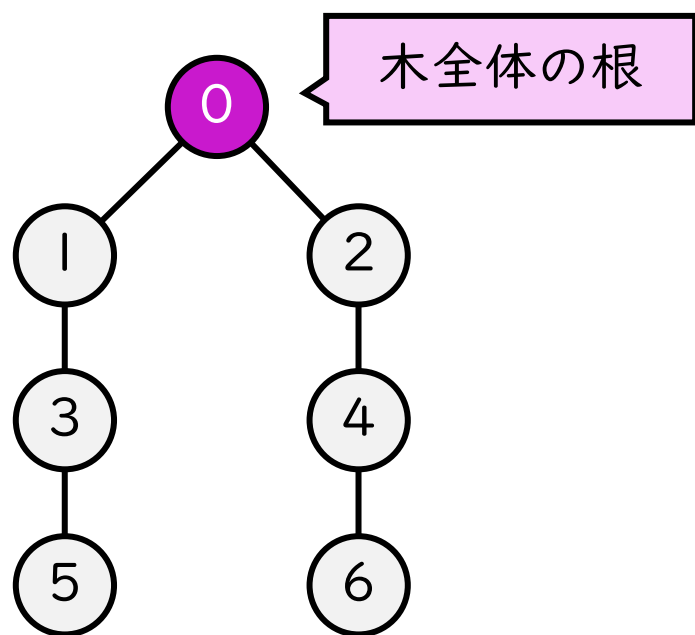
以下のようなものは絶対にあり得ない



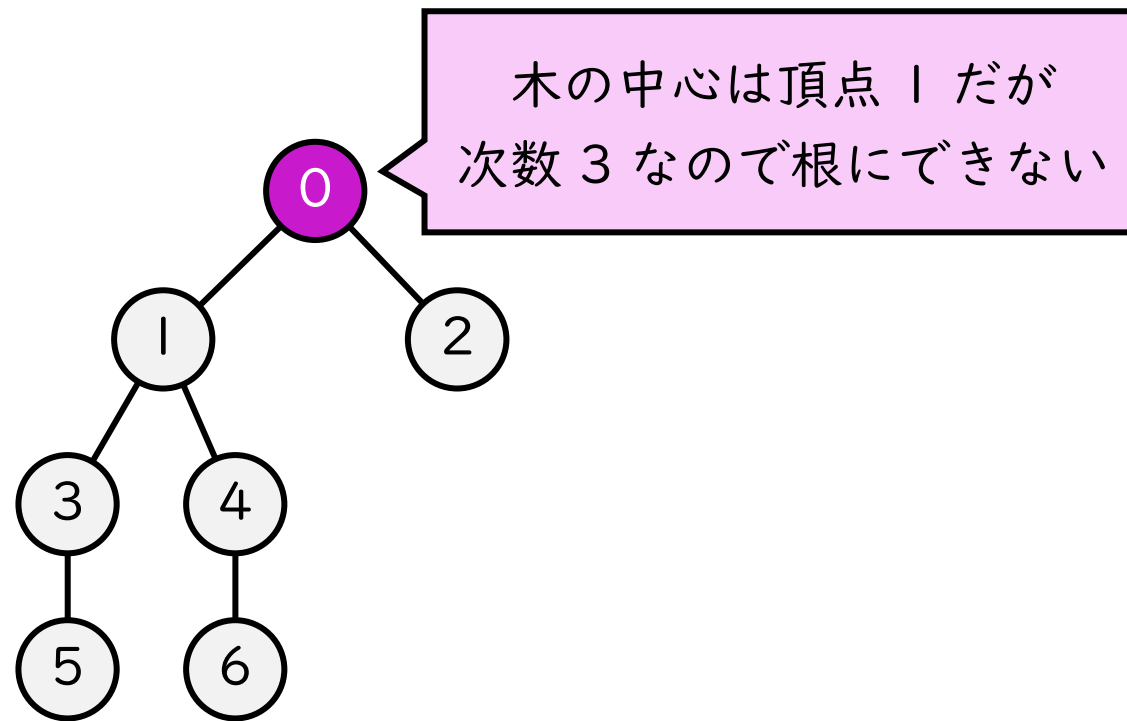
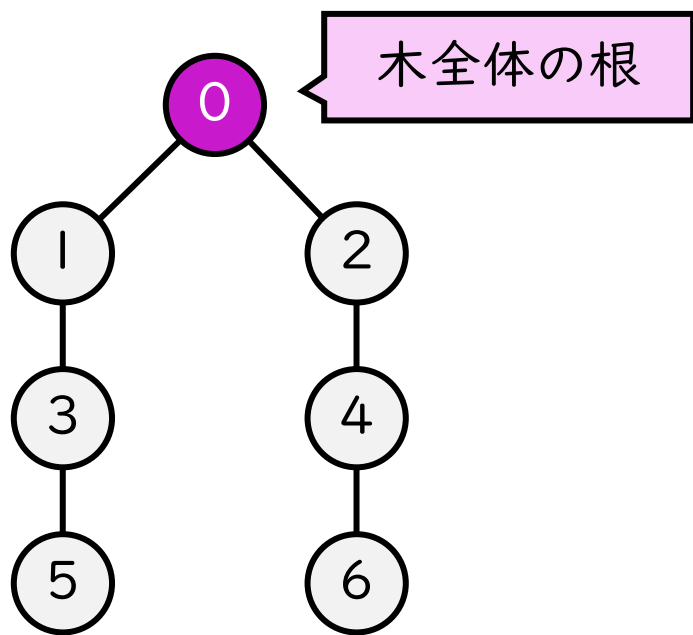
このようなものを排除すると...

676 通り ▶ 553 通り

できるだけ木の中心に近い次数 2 以上の頂点を
木全体の根にすることを考える



できるだけ木の中心に近い次数 2 以上の頂点を
木全体の根にすることを考える

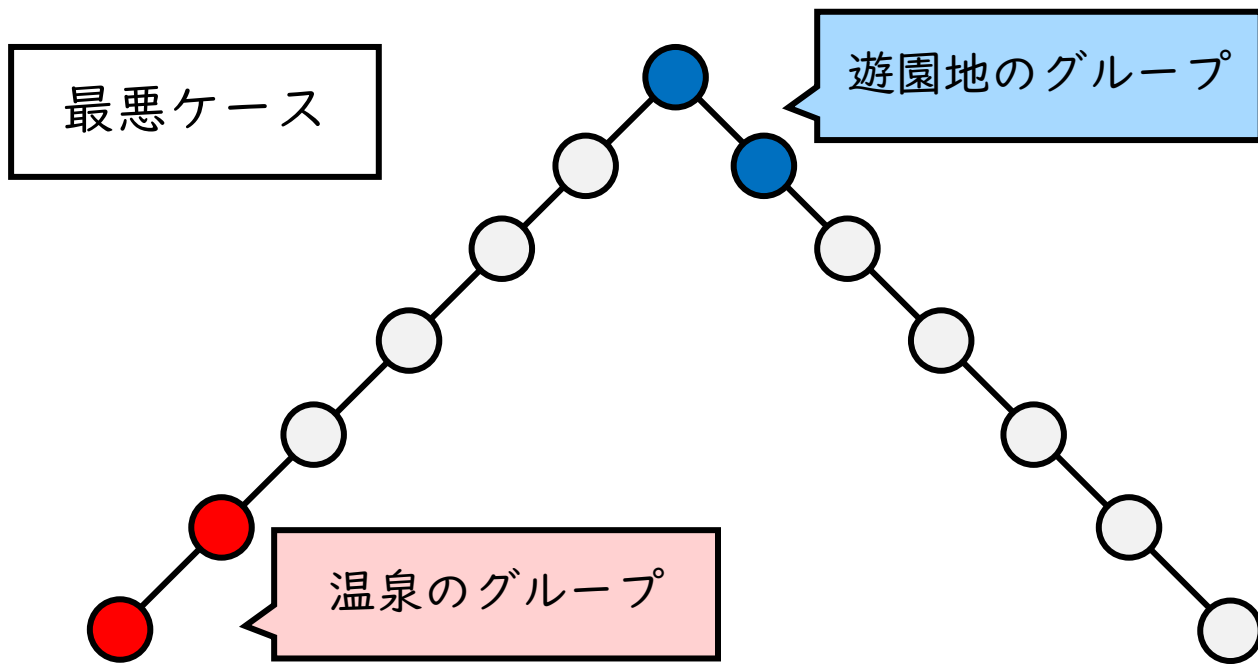


根を上手く設定すると...

端点 u, v の片方が「グループ内の根」ではないケースでは

$u - v$ 間距離

5000 以下



パターン数はどれくらいか？

u, v 両方がグループ内の根

➔ $38 \times 38 \times 10000$ 通り

そうでない場合

➔ $38 \times 515 \times 5000$ 通り※

合計 112,290,000 通り

※ 515 通りは、全体 553 通りから「端点がグループ内の根 (38 通り)」を引いた値



パターン数はどれくらいか？

u, v 両方がグループ内の根



$38 \times 38 \times 100000$ 通り

そうでない場合

26

ビット

ト

98

点

合計 112,290,000 通り

※ 515 通りは、全体 553 通りから「端点がグループ内の根 (38 通り)」を引いた値

実は頂点数を 14 まで増やしても

ID が $2N + 19$ を超えない

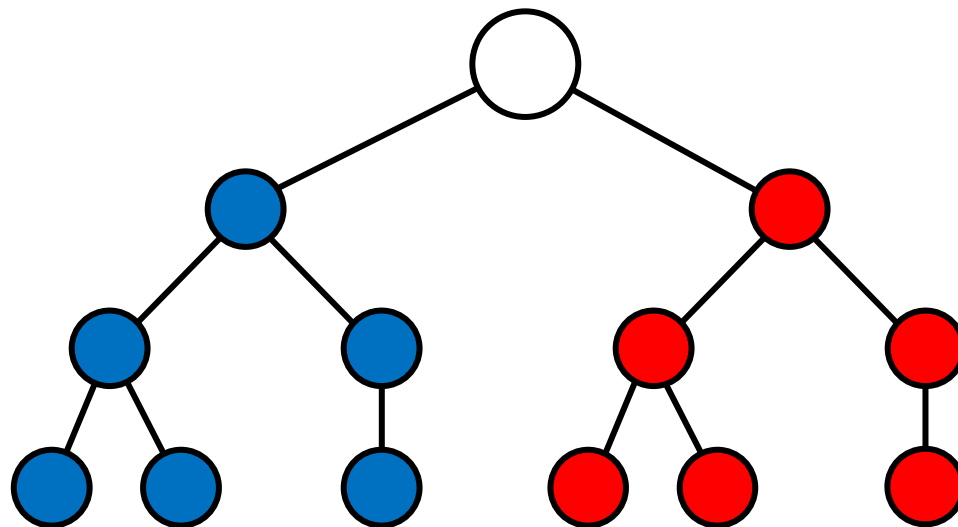


1 頂点追加して木の種類数を減らせないか？

重要な性質

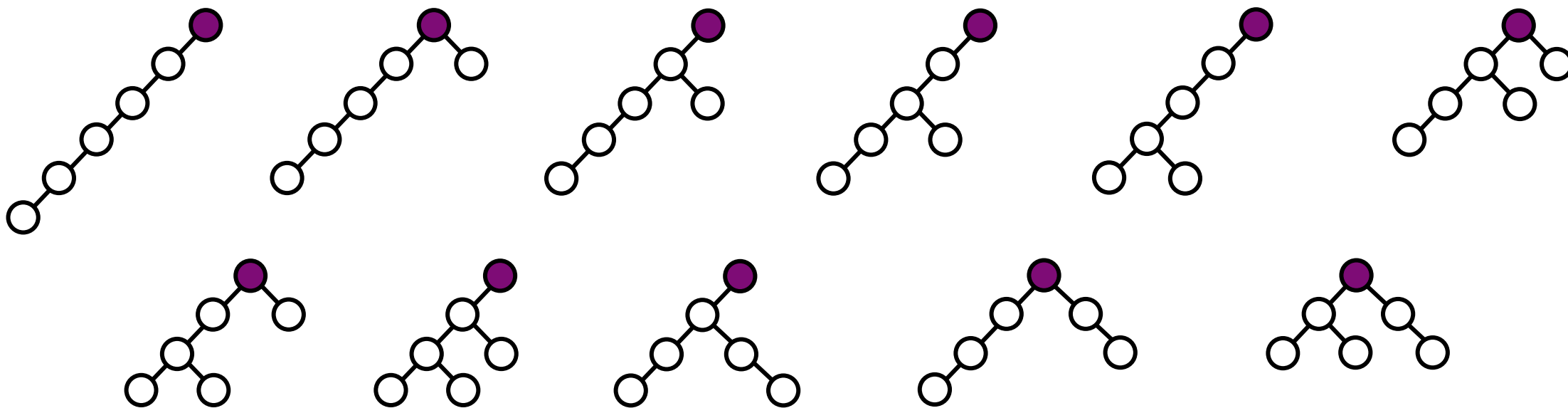
13 頂点の木の構成は

左部分木 6 + 中央 1 + 右部分木 6



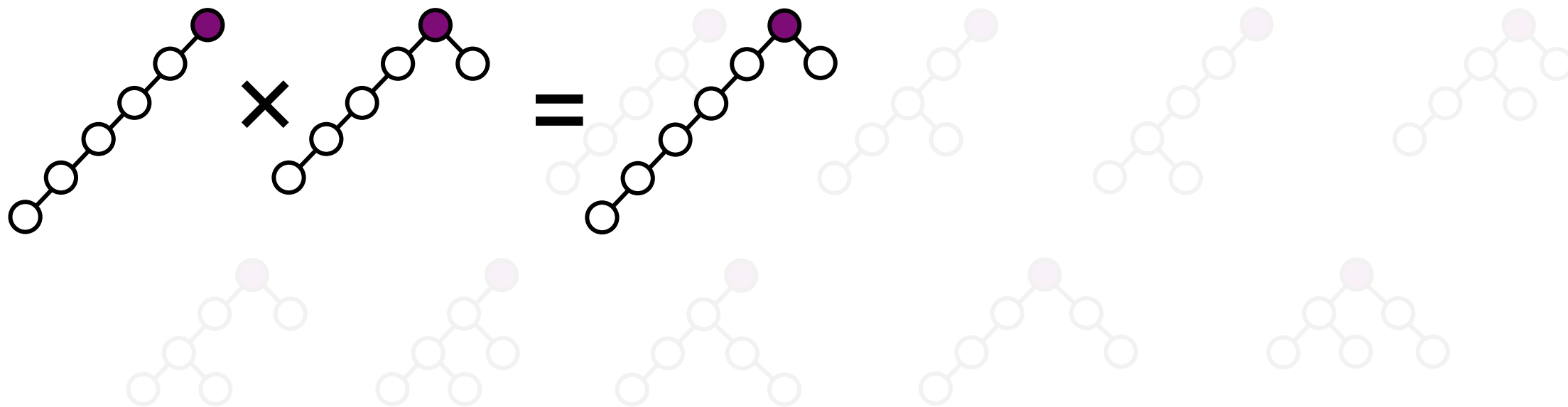
そこで、6 頂点の根付き二分木は全部で 11 個ある

※このことから、木の個数が $12 \times 11 \div 2 = 66$ 個と計算できる



しかしながら、頂点を 1 個追加すると 2 つの木を合成できる

※ サイズ 7 の二分木は、サイズ 6 の二分木を両方含む

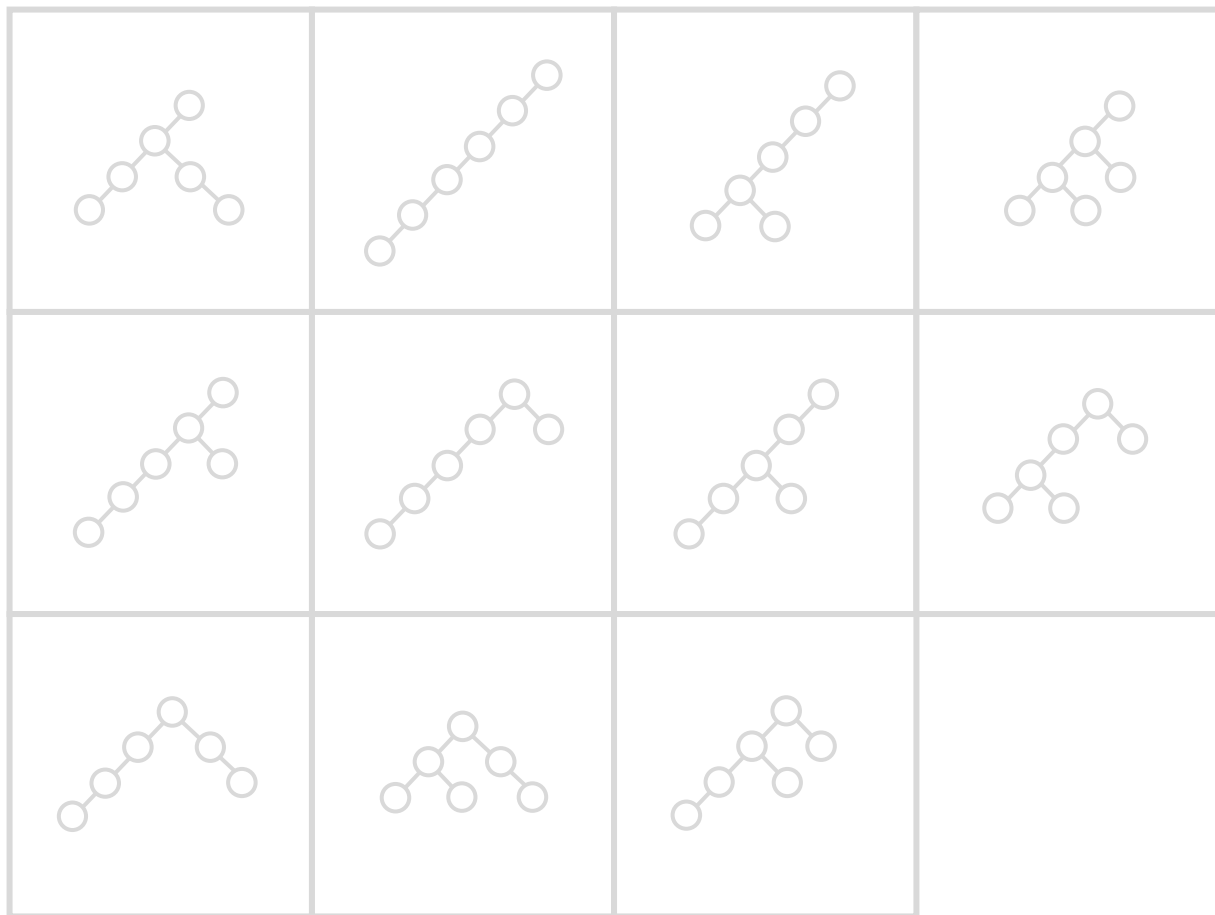


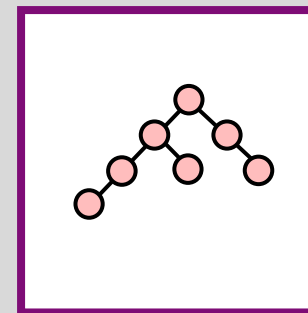
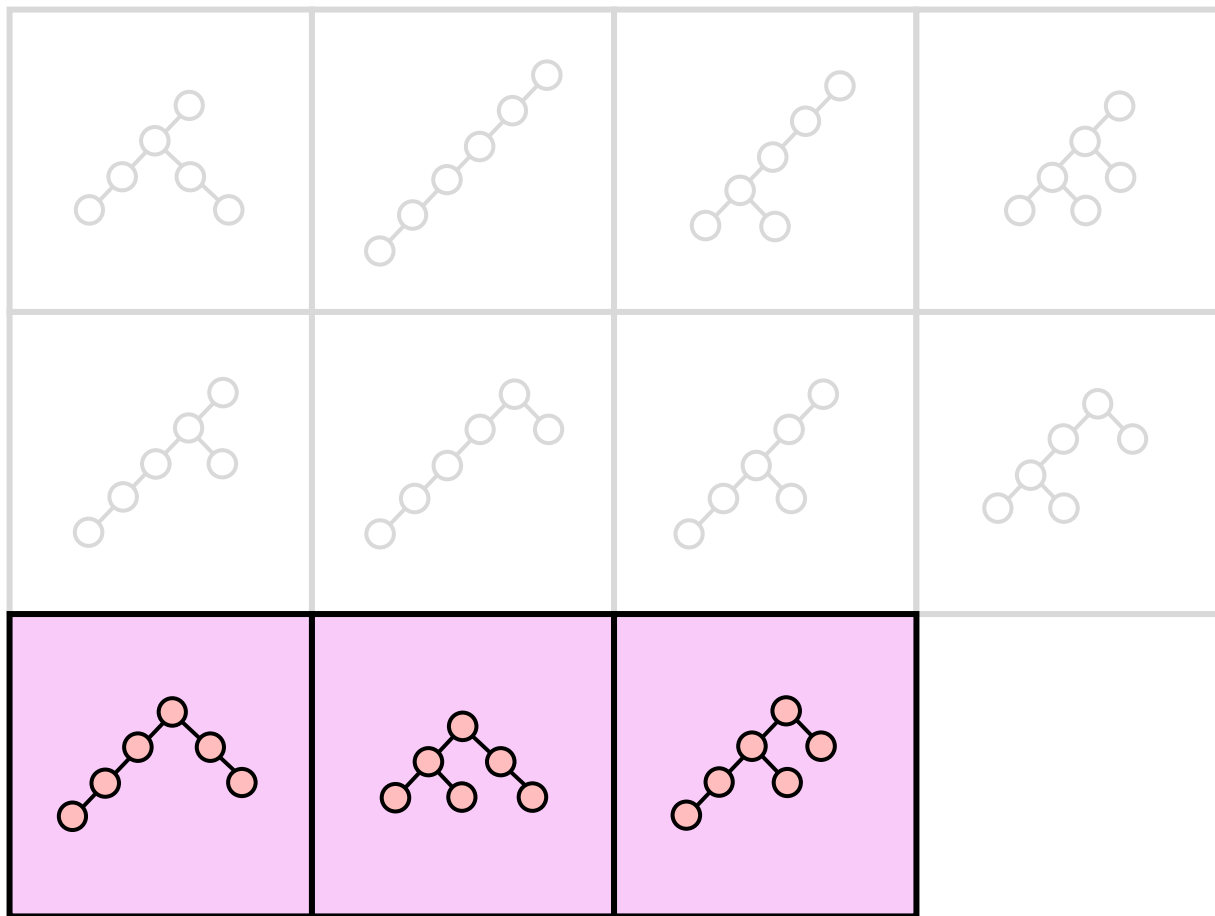
しかしながら、頂点を 1 個追加すると 2 つの木を合成できる

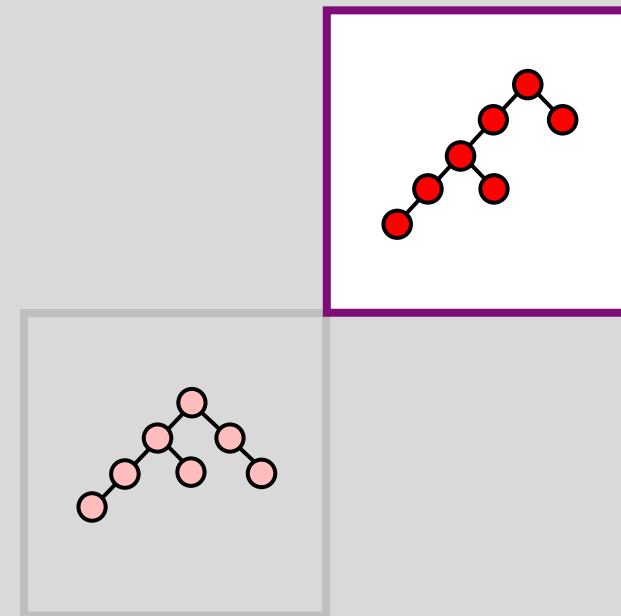
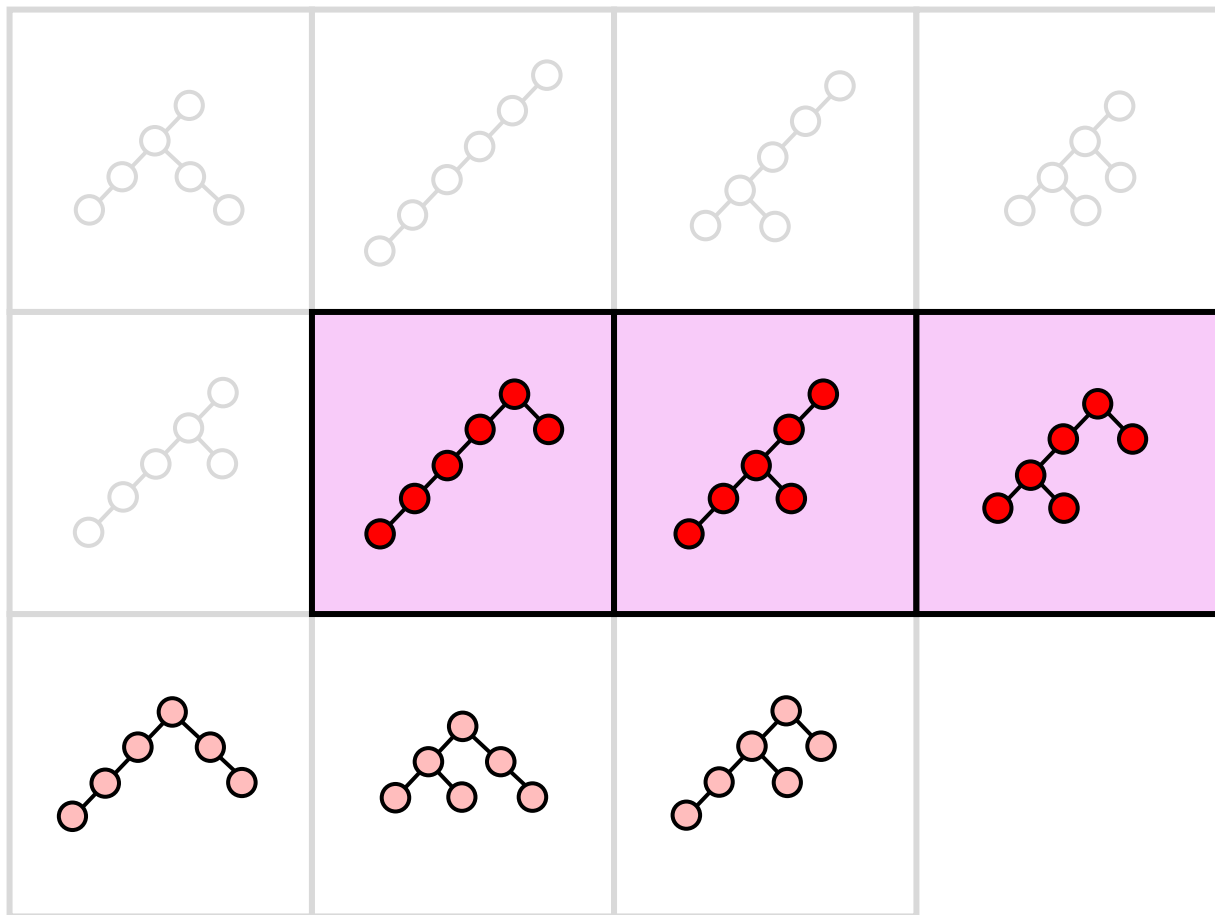
※ サイズ 7 の二分木は、サイズ 6 の二分木を両方含む

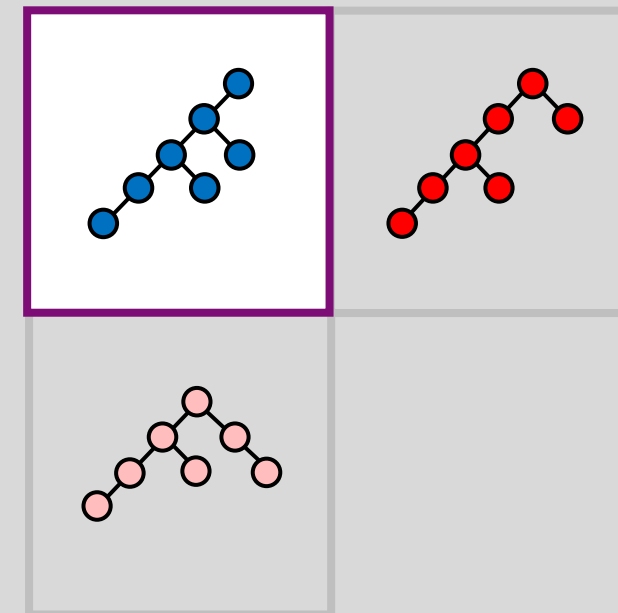
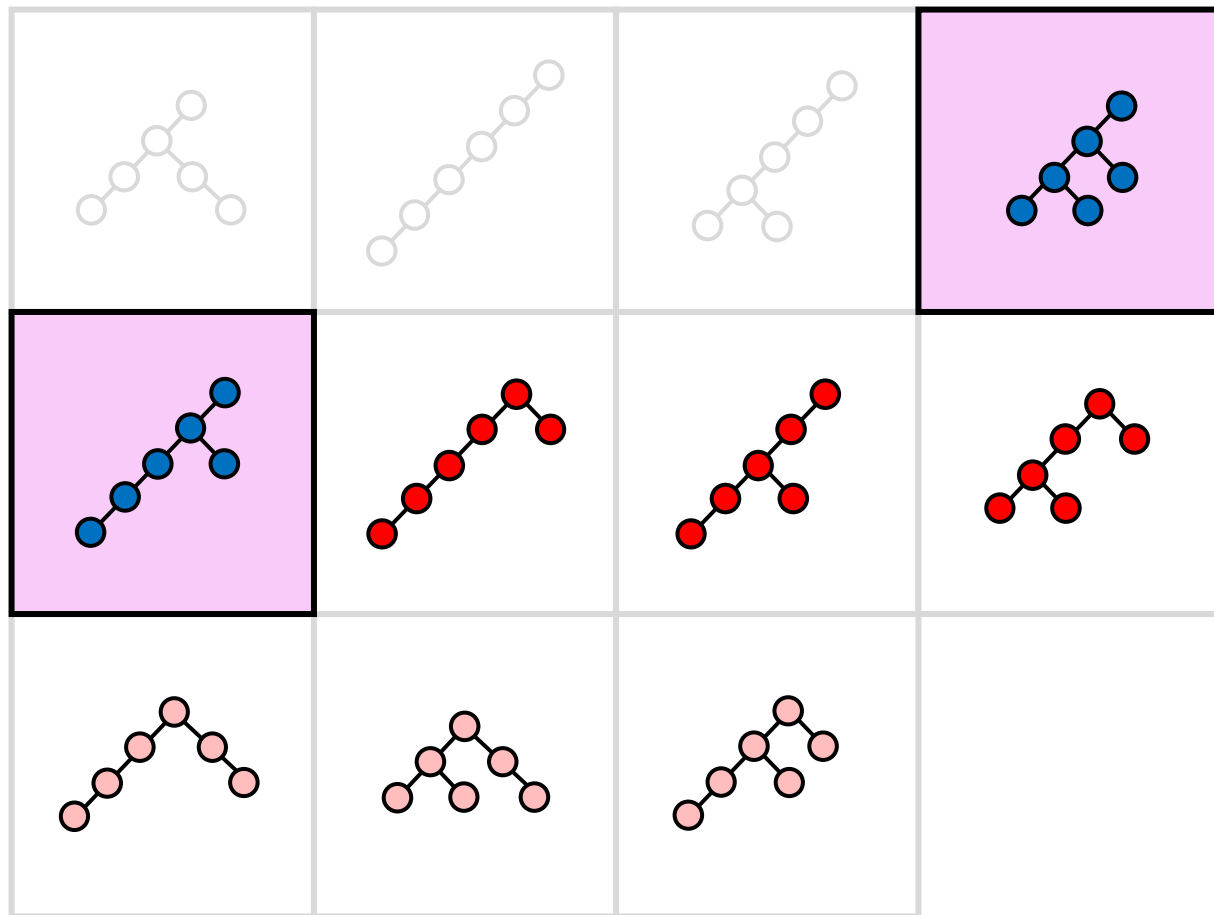
どう合成すれば

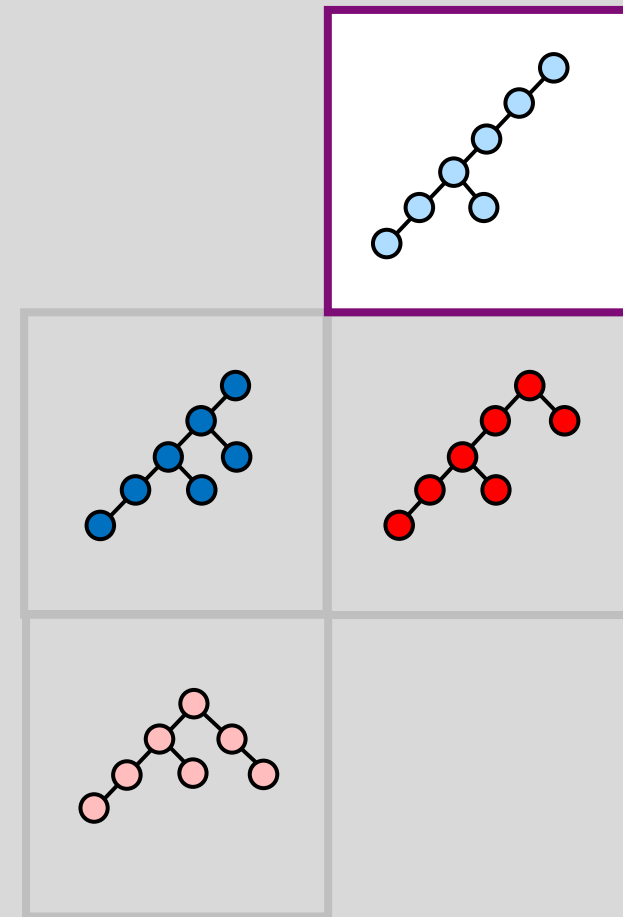
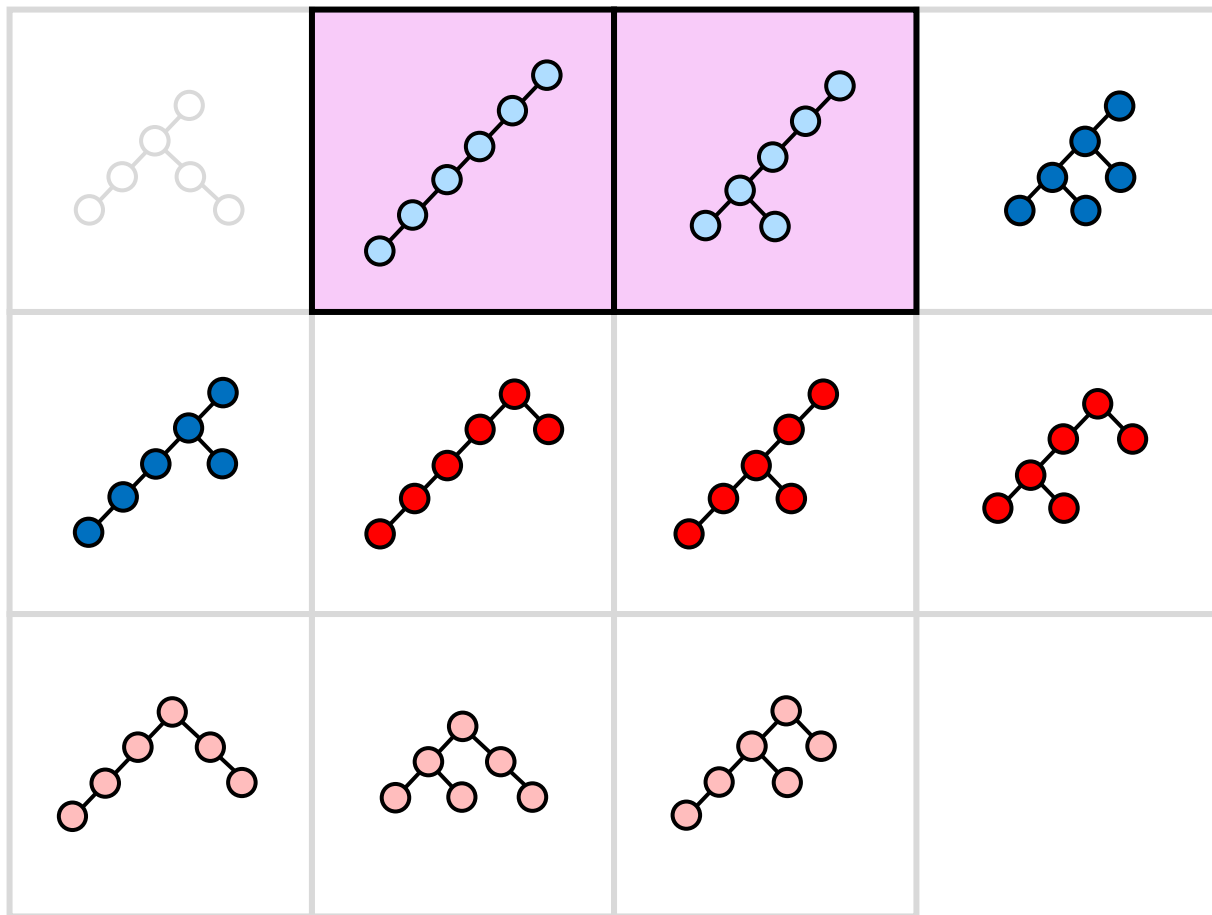
木の個数が少なくなるのか？

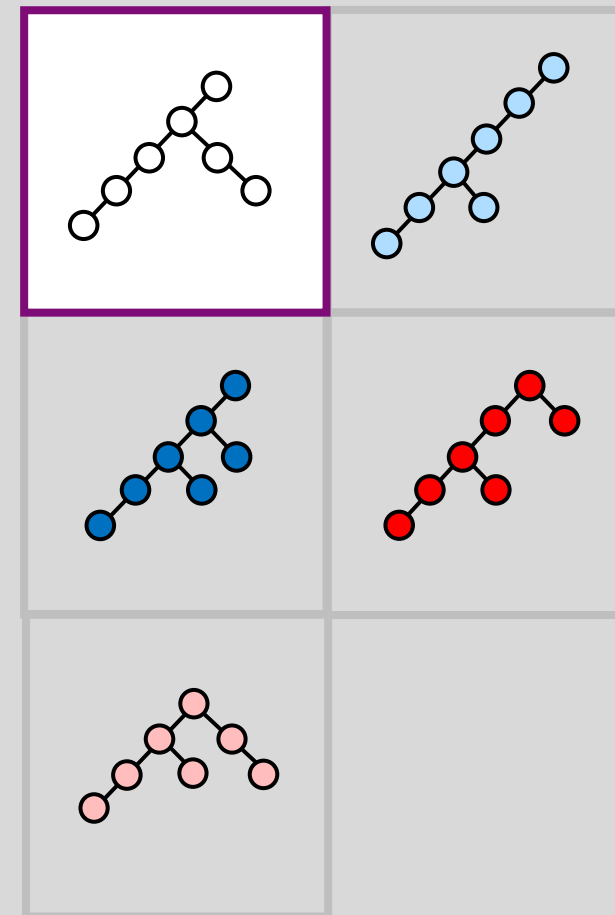
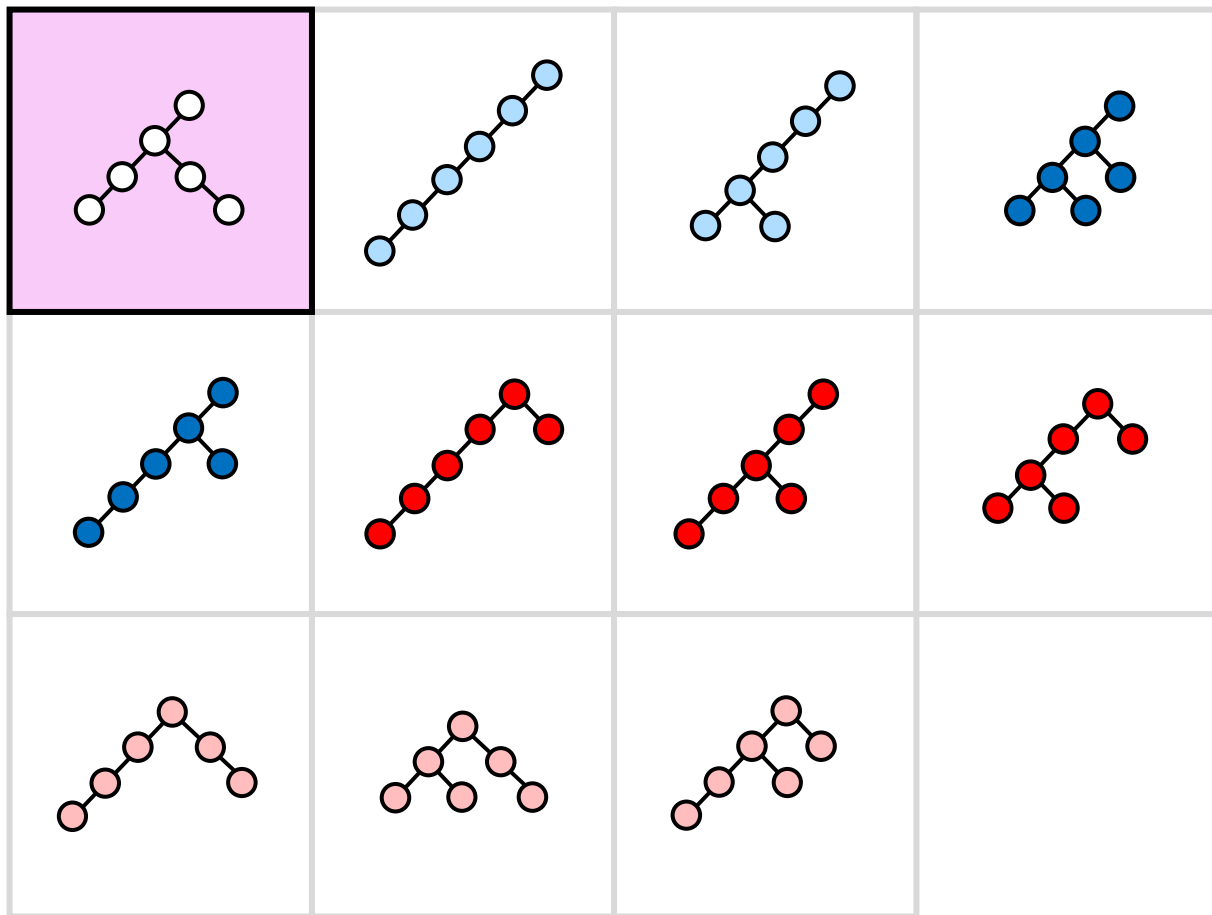


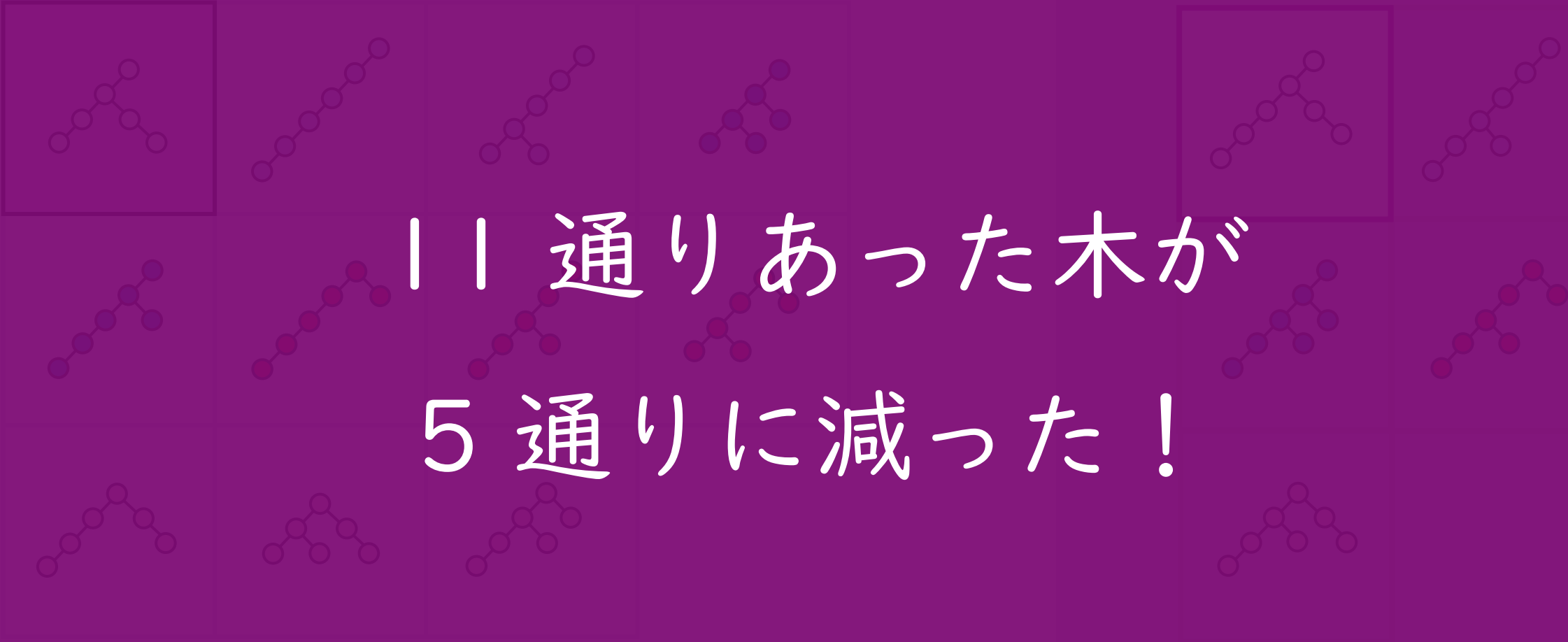






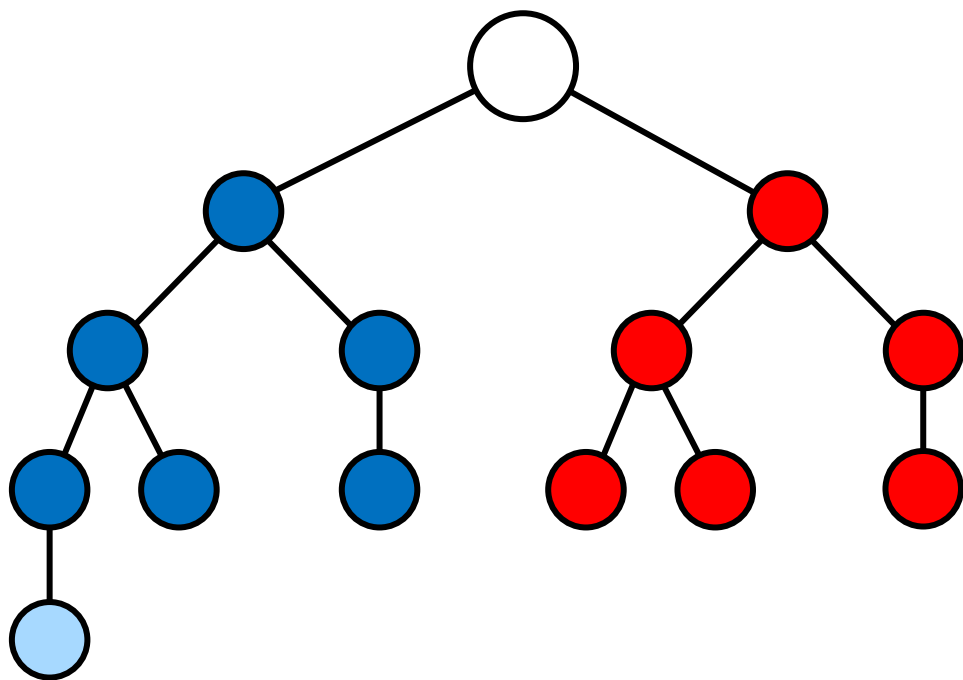






11通りあった木が
5通りに減った！

左側の部分木に 1 頂点追加することを考えると...

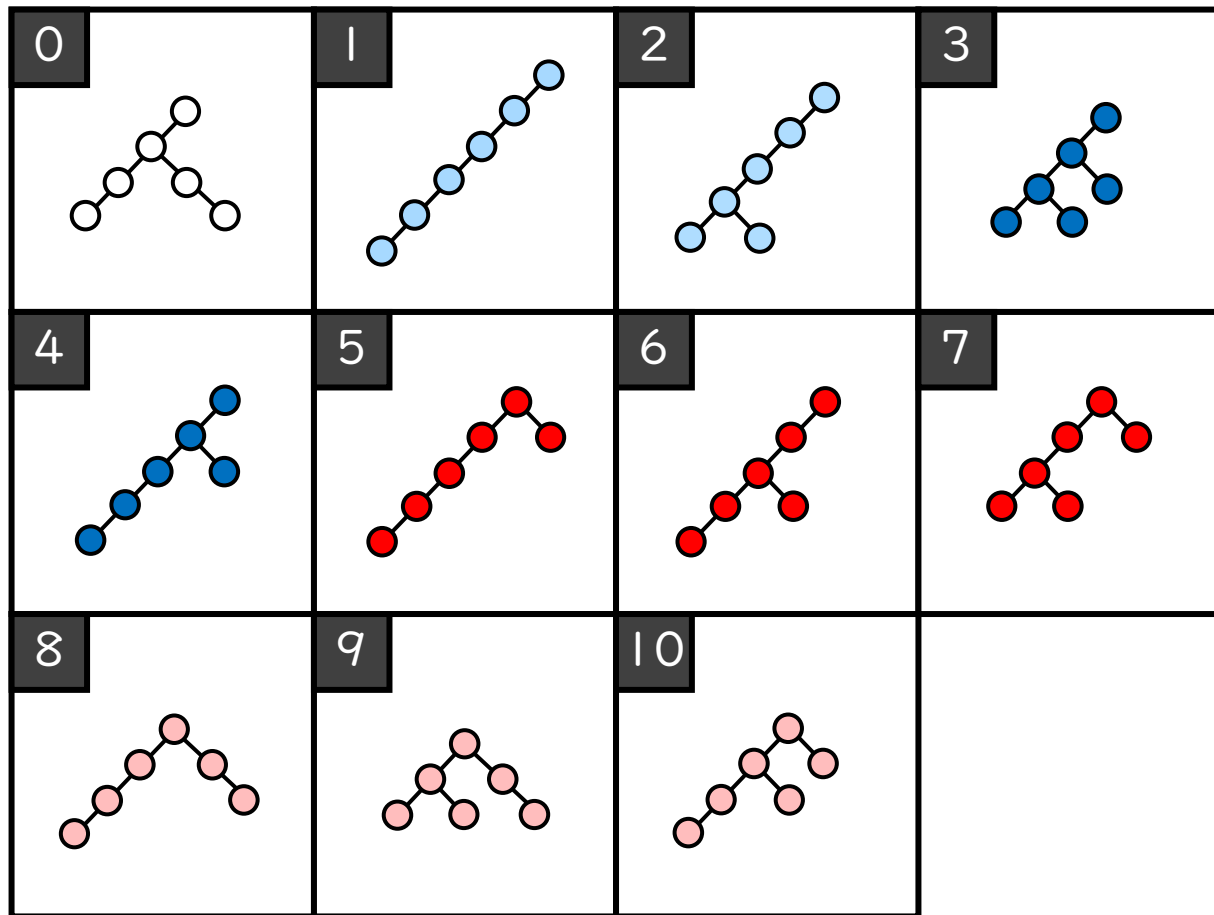


元々 66 通りあったのが

$$\underline{5} \times \underline{11} = 55 \text{ 通り}$$

左部分木 右部分木

に減ったように思える



しかし実際はもっと減る

白色の二分木から順に 0, 1, 2, ..., 10
の番号を付けて、番号の大きい方に 1
頂点を追加すると...



木の数は **28** 通り

$$\text{※ } 5+4+4+3+3+2+2+2+1+1+1=28$$

この方針で距離分布を送ると、一部重複が出るのでパターン数はもっと減る（次の通り）。

端点が根 22通り

全部で 304通り



$$22 \times (304 + 22) \times 5000 \\ = 3586 \text{ 万通り}$$

2^{25} は約 3355 万なので
惜しくも 25 ビットになってしまう…



この方針で距離分布を送ると、一部重複が出るのでパターン数はもっと減る（次の通り）。

25

ビット



99

点

端点がある

22通り

全部で

304通り

$$2^9 \times (304 + 12) \times 5000$$

$$= 3586 \text{ 万通り}$$

2^{25} は約 3355 万なので

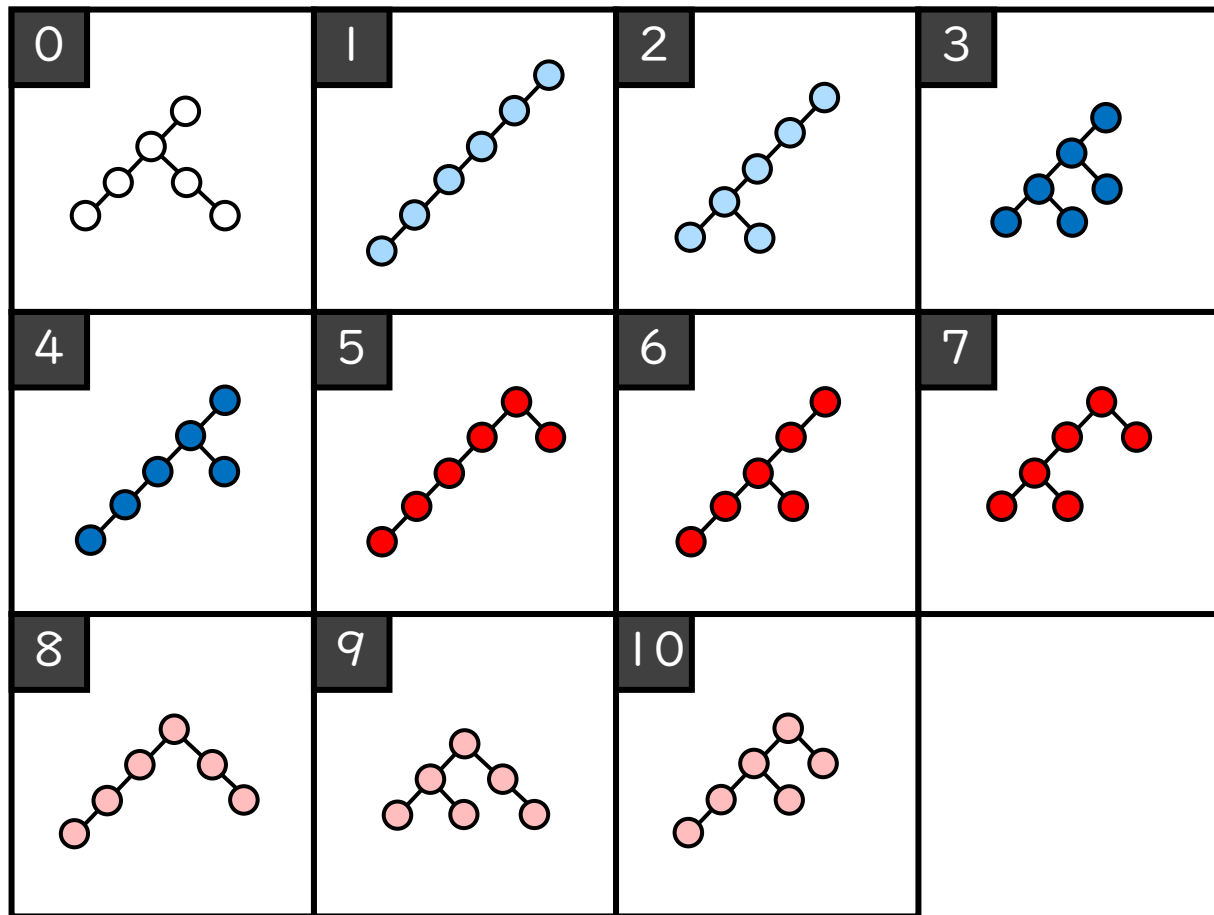
惜しくも 25 ビットになってしまう…

最後の一押し

番号の振り方を変えよう！

6

満点解法

$$\begin{array}{r} 213 \\ \hline 222 \end{array}$$


端点が根 22通り

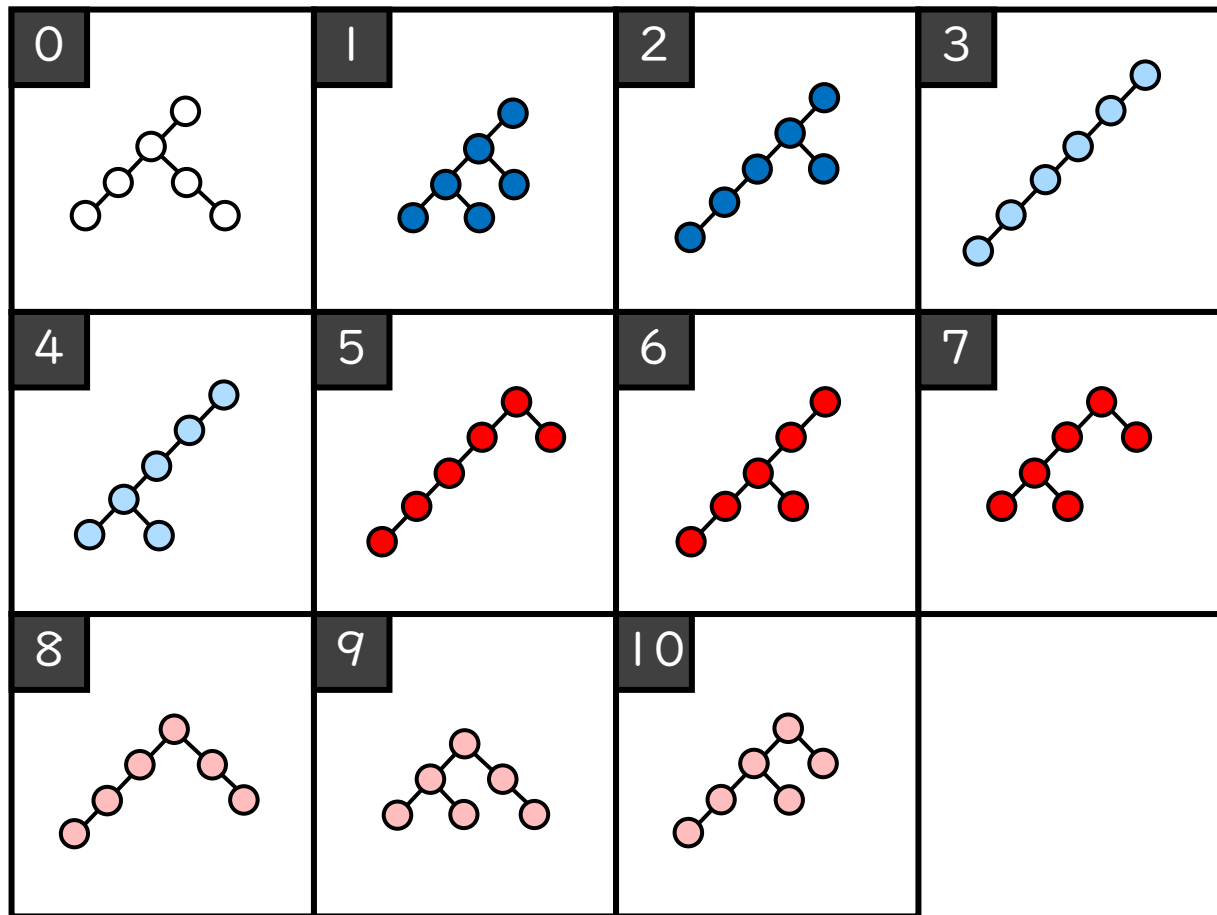
全部で 304通り



識別すべきパターン数は

3586万通り

25ビット



端点が根 22通り

全部で 296通り



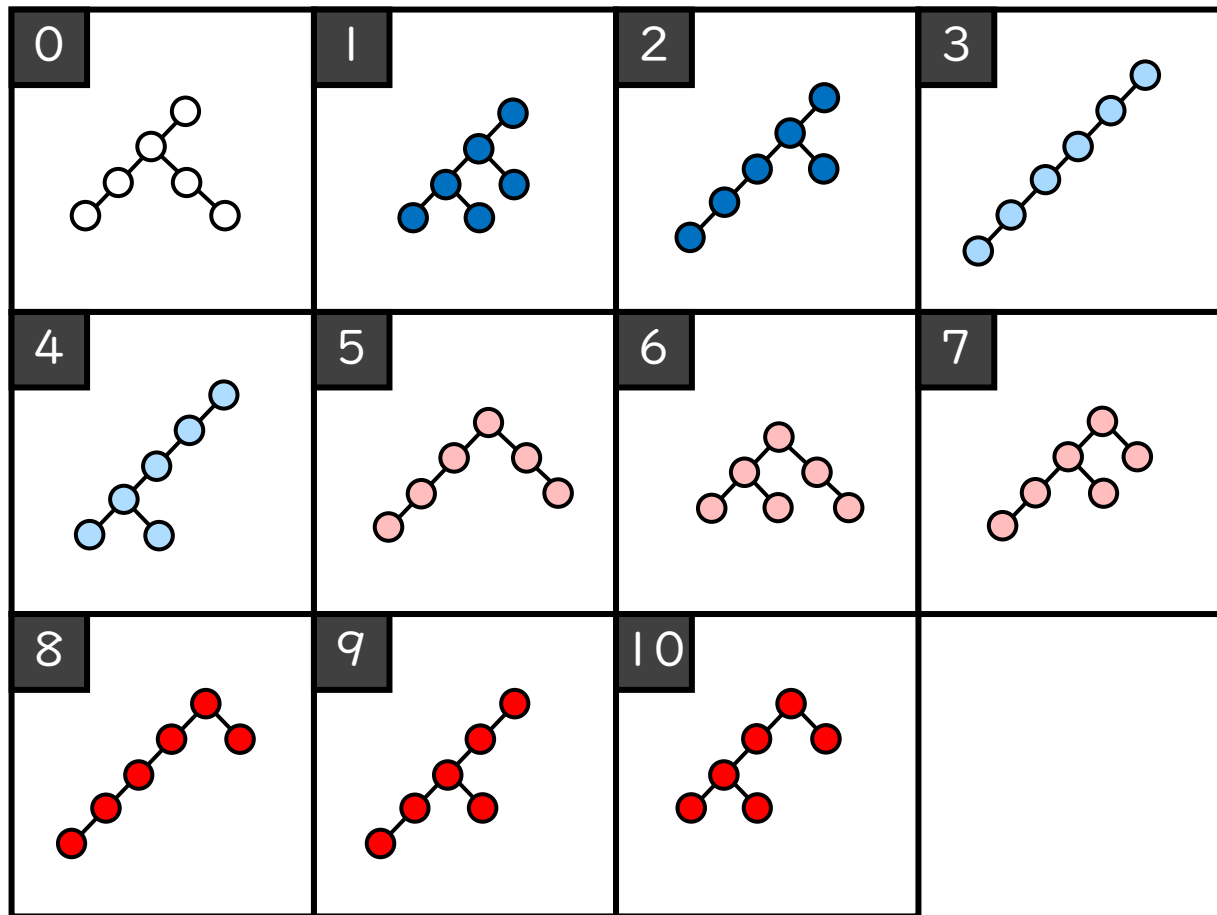
識別すべきパターン数は

3498万通り

25ビット

6

満点解法

$$\frac{215}{222}$$


端点が根 21通り

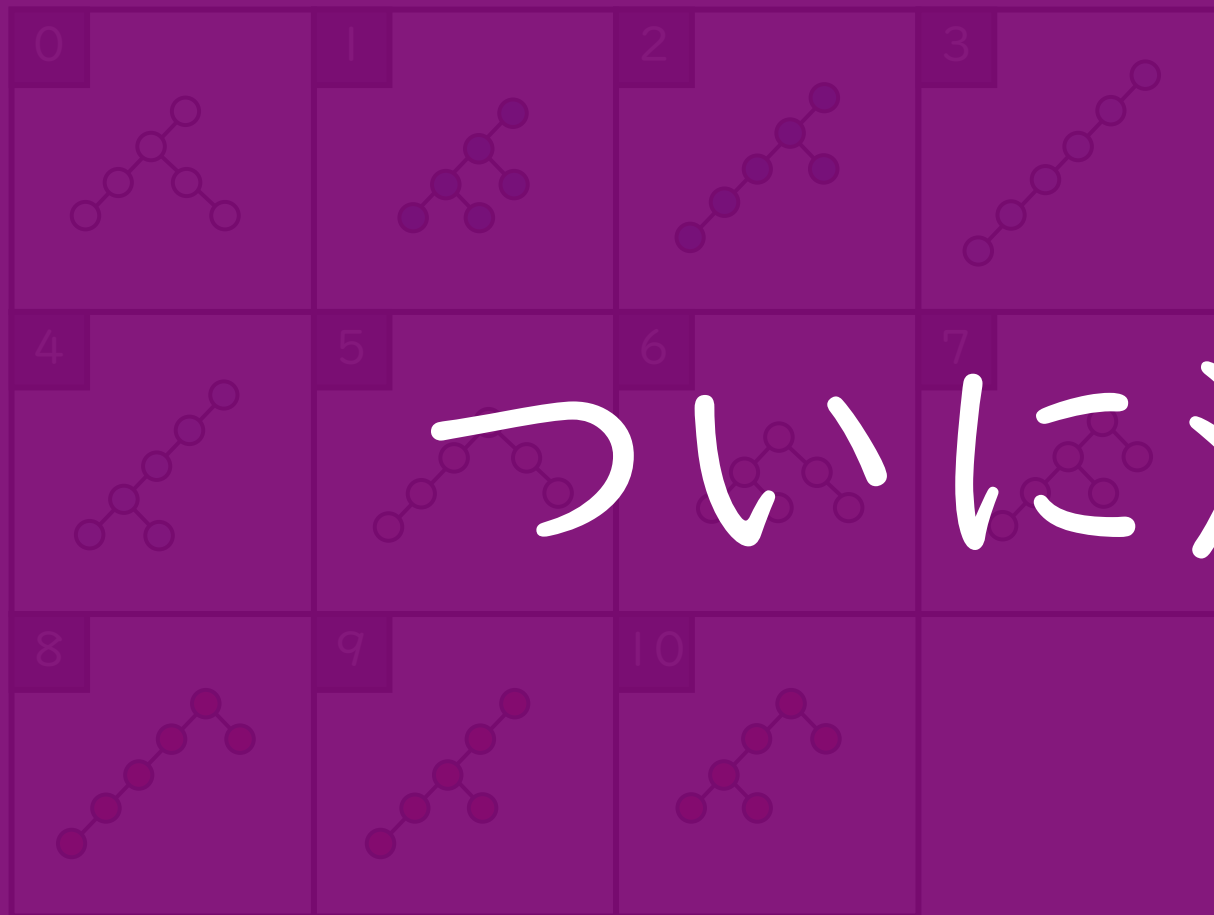
全部で 290通り



識別すべきパターン数は

3265万通り

24ビット



ついに満点!

端点が根

21通り

全部で

290通り

識別すべきパターン数は

3265万通り

24ビット

解法のみまとめ

- 木を連結なグループに分解することを考える
 - 下から貪欲に切っていくことで、根以外の全グループの頂点数を 7~13 にできる
 - グループ数は 1429 以下なので、遊園地・温泉の順序が関係ないことを使えば、Benjamin は 20 ビットで送れる



ここまで基礎点 59 点

さらなる改善として、次のようなものが考えられる

ステップ	改善の内容	ビット数	得点
1	N 頂点の木を $2N$ ビットで送る	70	67 点
2	木が高々「カタラン数」通りであることを使う	60	72 点
3	次数 ≤ 3 などの条件を追加してパターン数を減らす	44	80 点
4	同型な木を同一視してパターン数を減らす	40	84 点
5	接点 u, v のうち片方しか送らなくて良い	36	88 点
6	すべてのグループを 13 頂点にする	32	92 点
7	4 つの要素をすべてまとめる	30	94 点

さらなる改善として、次のようなものが考えられる

ステップ	改善の内容	ビット数	得点
8	1 ~ L ビット全部利用できることに気づく	29	95 点
9	木の代わりに距離分布を送る	27	97 点
10	根を「木の中心」付近にする	26	98 点
11	上手く追加して各グループ 14 頂点にする	24	100 点

- この問題は、少なくとも昨年度までのどの JOI 過去問よりも難易度が高いと思います
- 満点を取るのは非常に難しいので、**部分点**を積極的に狙う、ほかの問題に手を付けるなどの戦略が有効です

