データ構造とアルゴリズム
7. 平衡木（AVL木）

http://numata.designed.jp/dg/

平衡木

平衡木 (Balanced Tree)

- 挿入・削除を行う度に木の形を整えて、
  常に探索が $O(\lg n)$ で行えるようにする木。
- 整形に探索以上のコストをかけては
  元も子もないので、整形のコストは
  $O(\lg n)$ 以下でなければいけない。

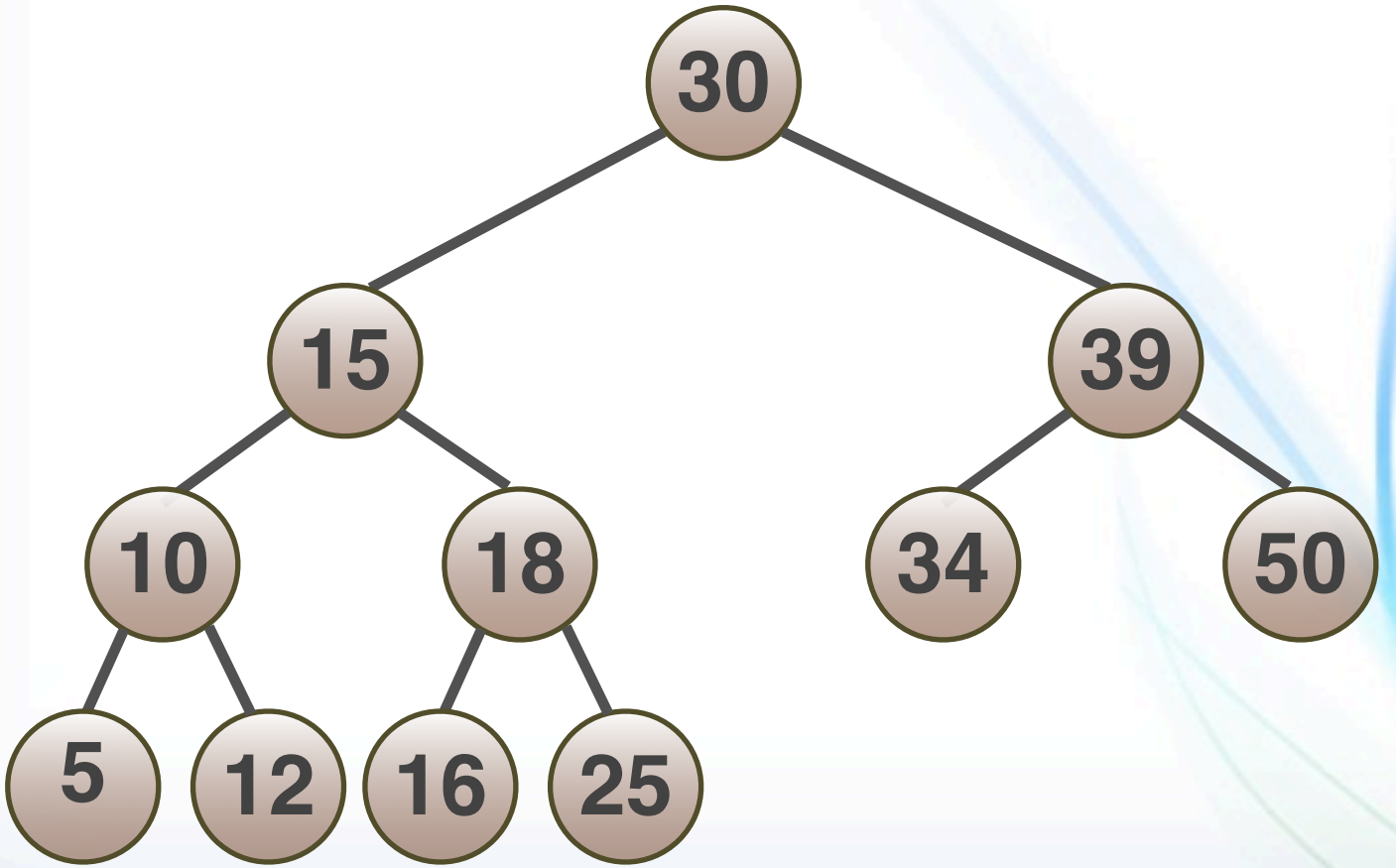
平衡木の種類

- AVL木
- B木
- 2-3木
- 2-3-4木
- 赤黒木
- スプレー木

AVL木

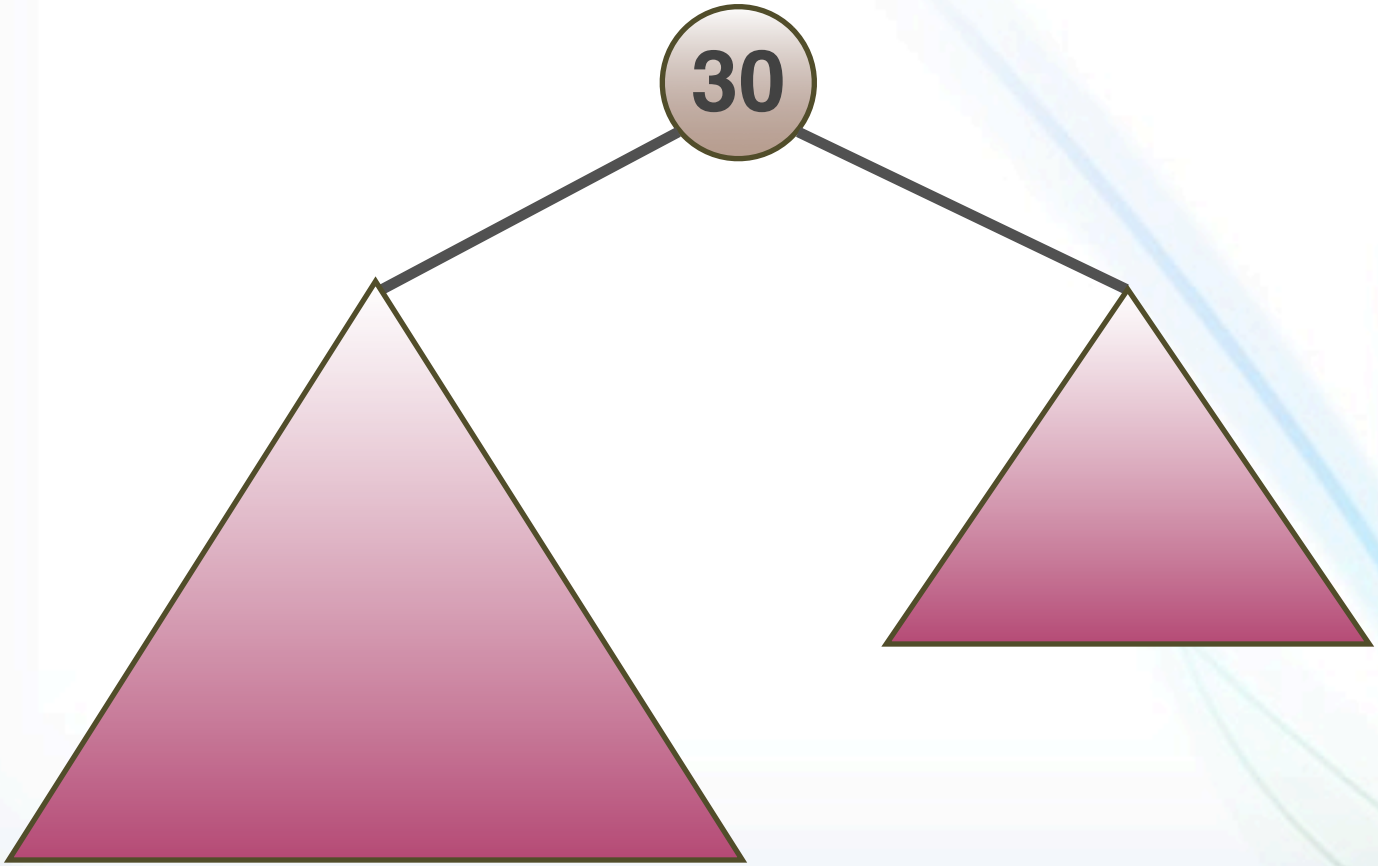
予備知識：部分木の表現

簡単のために、部分木を三角で省略する。



予備知識：部分木の表現

簡単のために、部分木を三角で省略する。

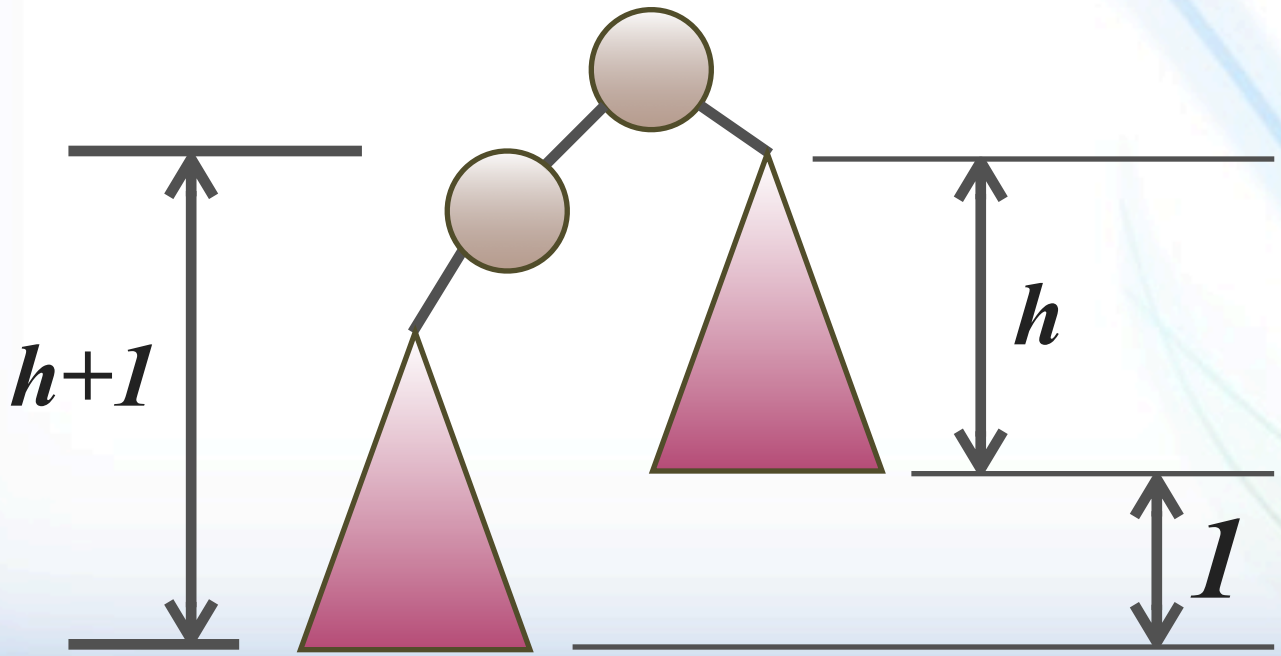


AVL木 (AVL Tree)

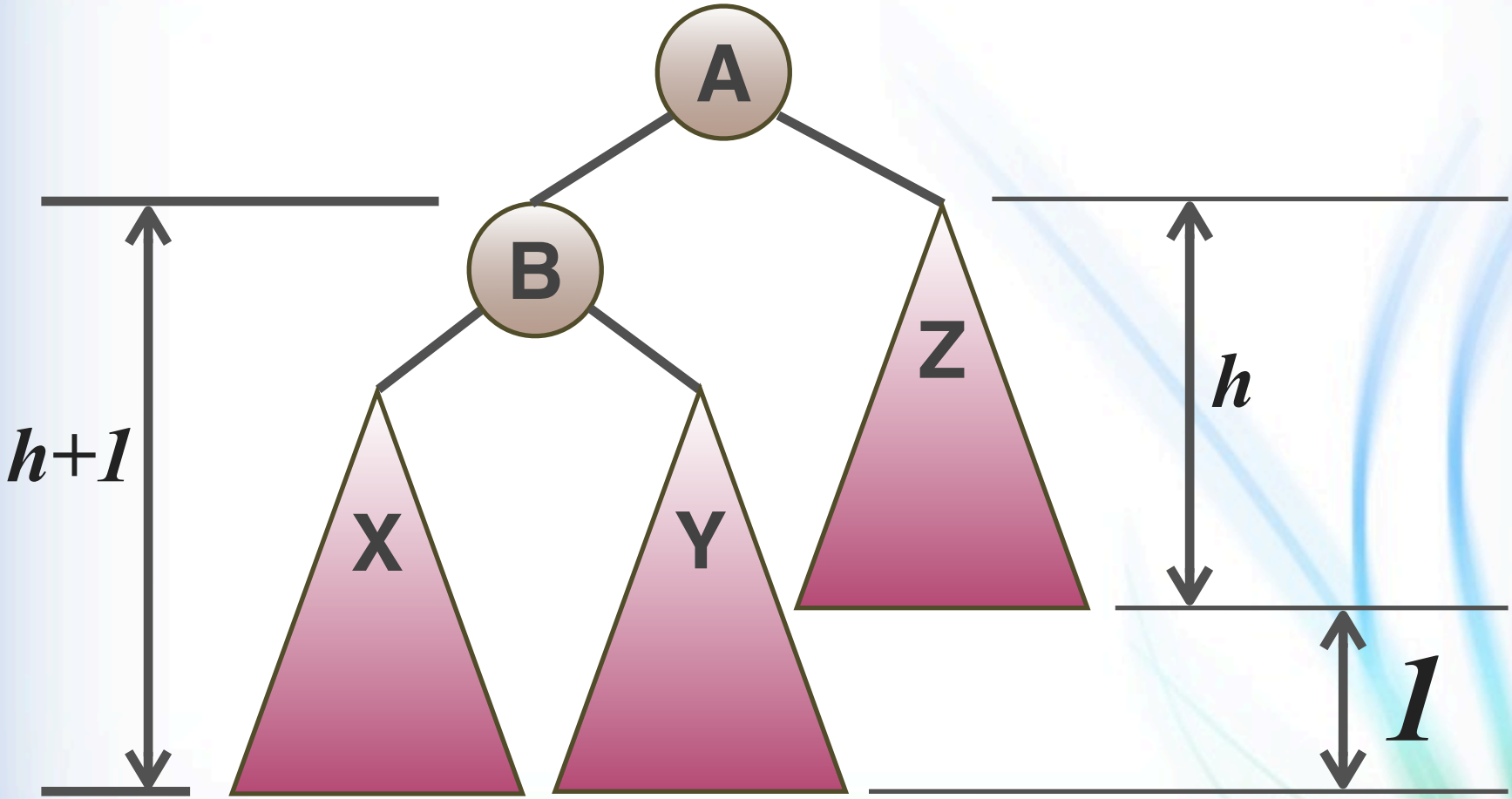
1962年、Adel'son-Vel'skii と Landisが考案。

制約条件

- 二分探索木（「左は小さく、右は大きい」）
- 左部分木の高さと右部分木の高さの差が１以内であること。

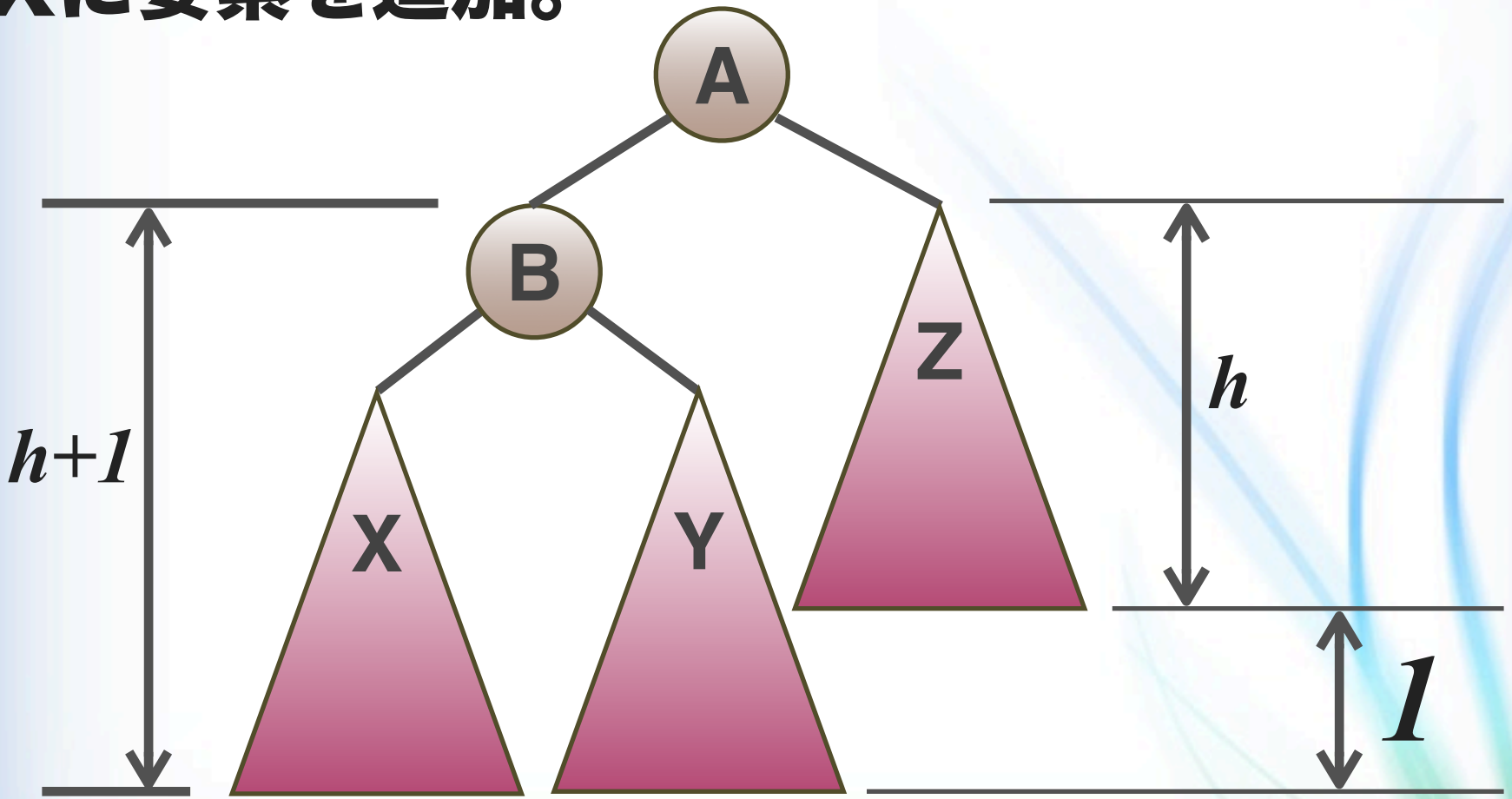


AVL木の基本形（平衡状態）



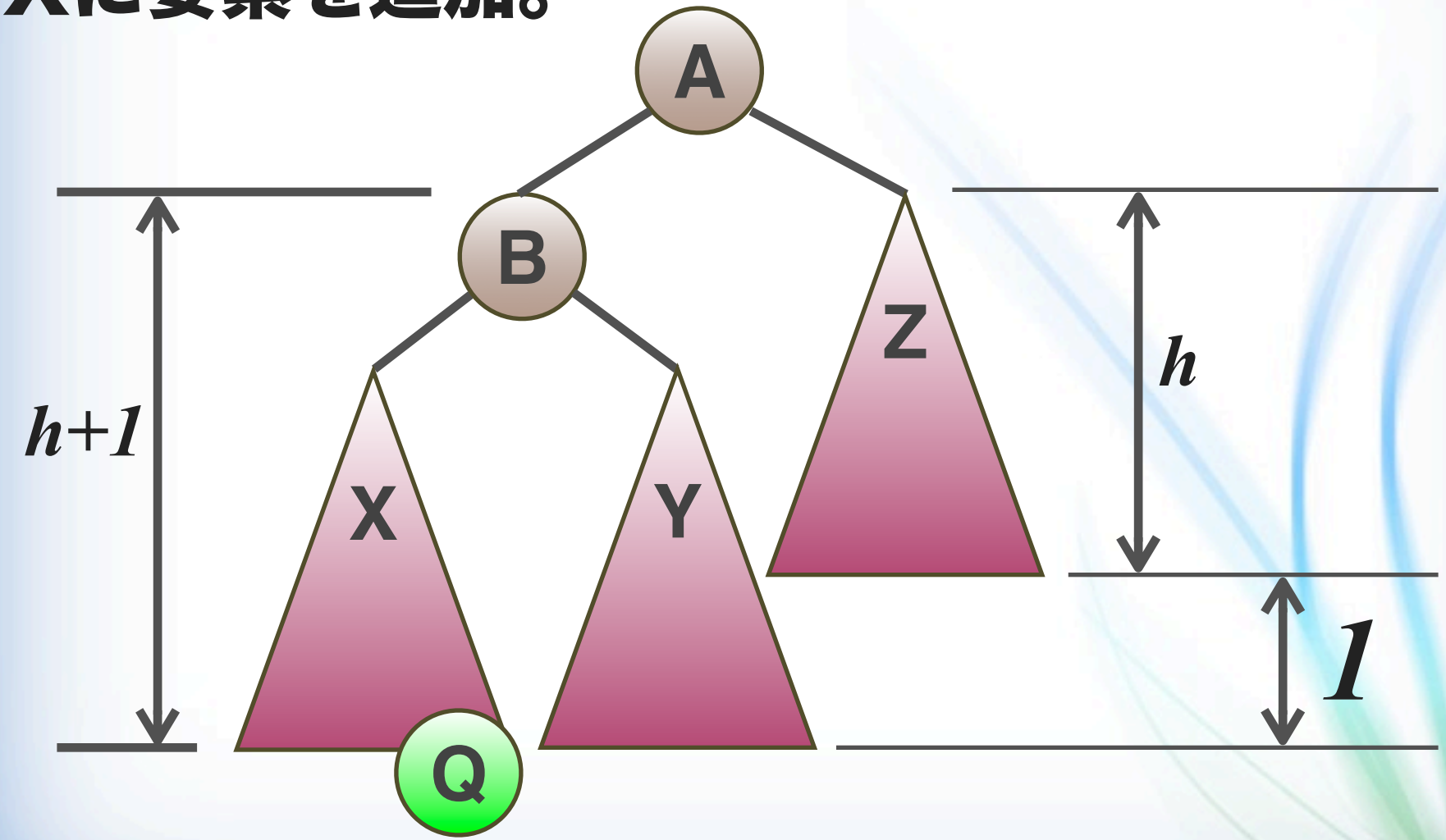
AVL木の追加処理（左側への追加）

Xに要素を追加。



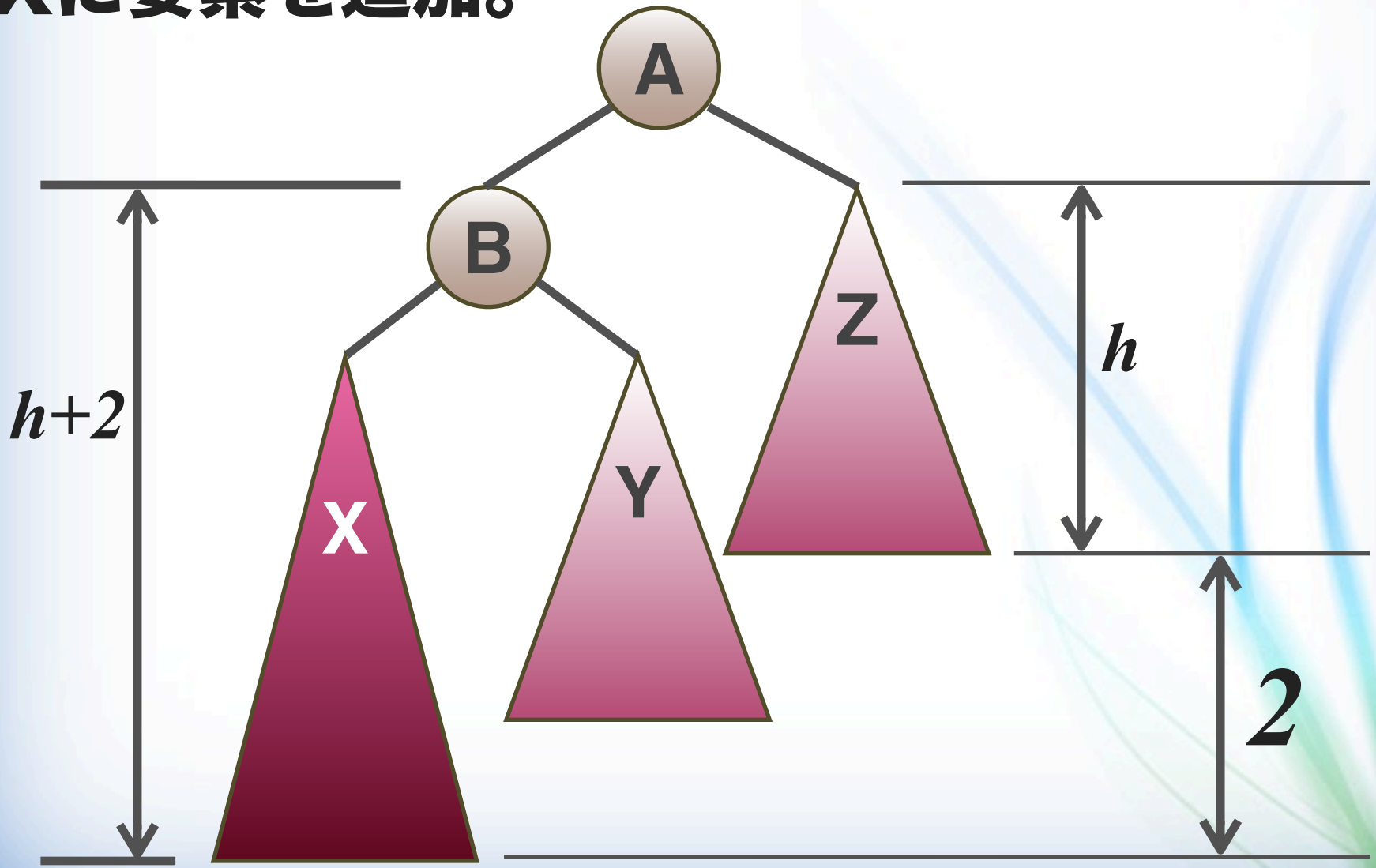
AVL木の追加処理（左側への追加）

Xに要素を追加。



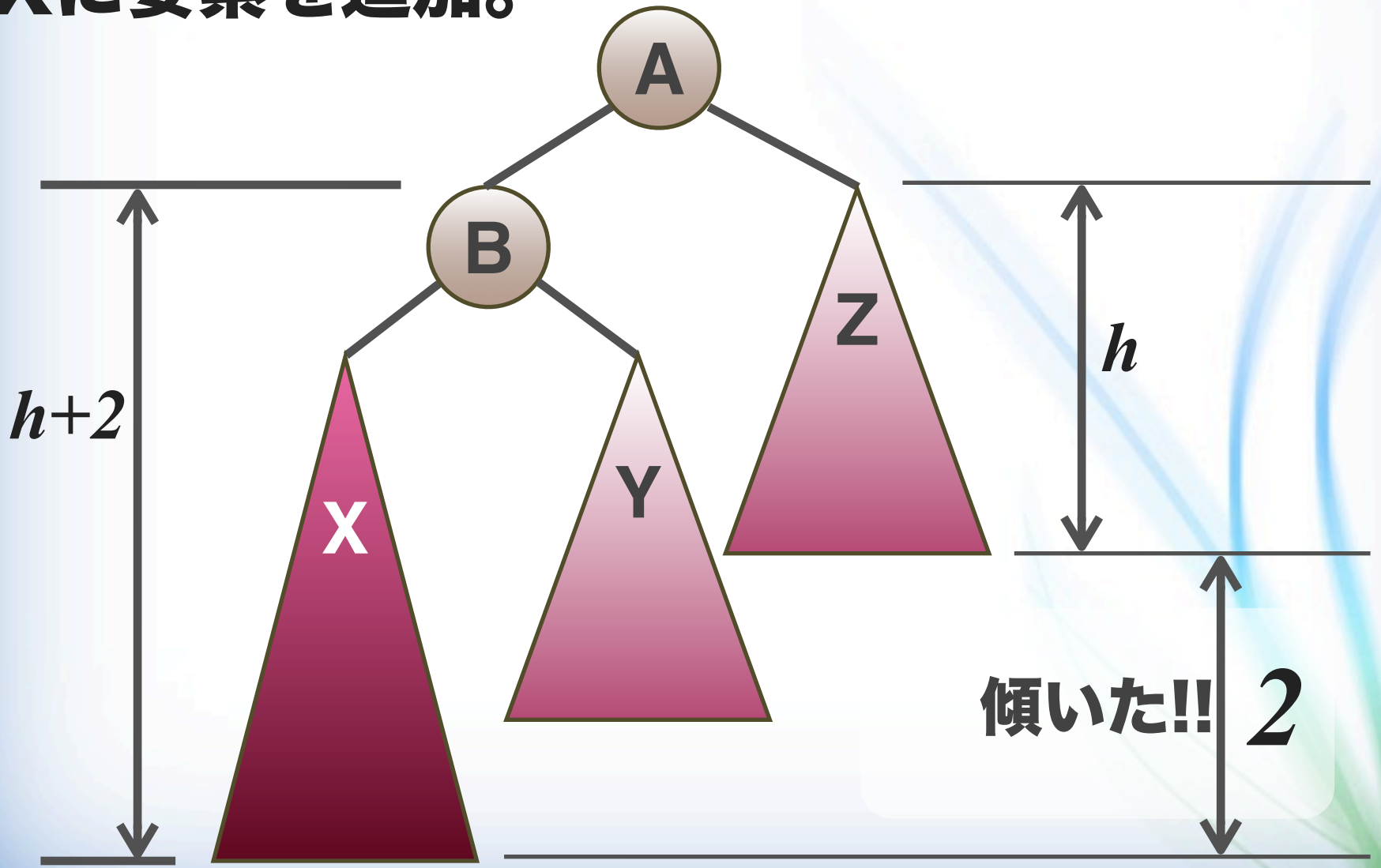
AVL木の追加処理（左側への追加）

Xに要素を追加。



AVL木の追加処理（左側への追加）

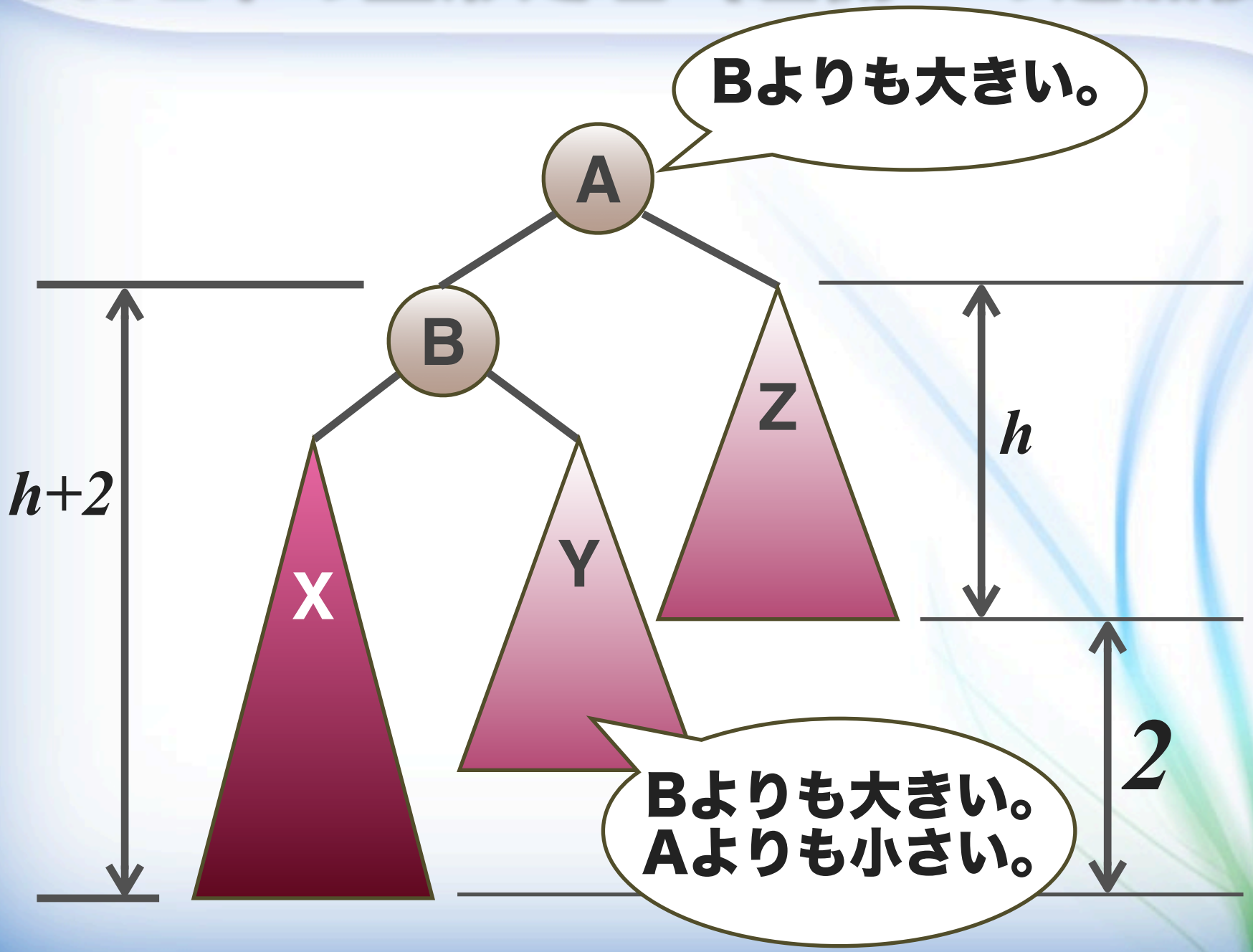
Xに要素を追加。



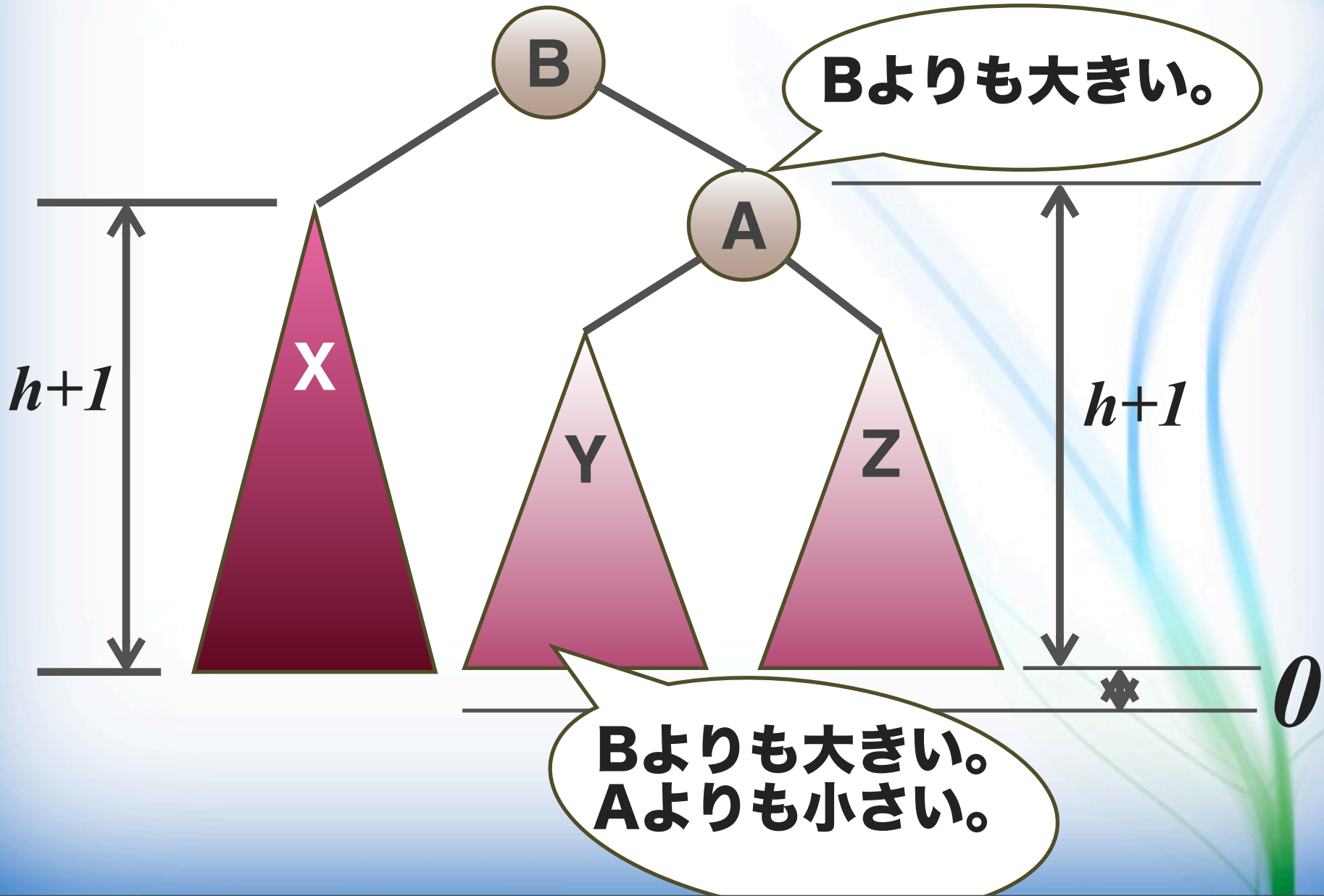
AVL木の追加処理（左側への追加）

Xに要素を追加。

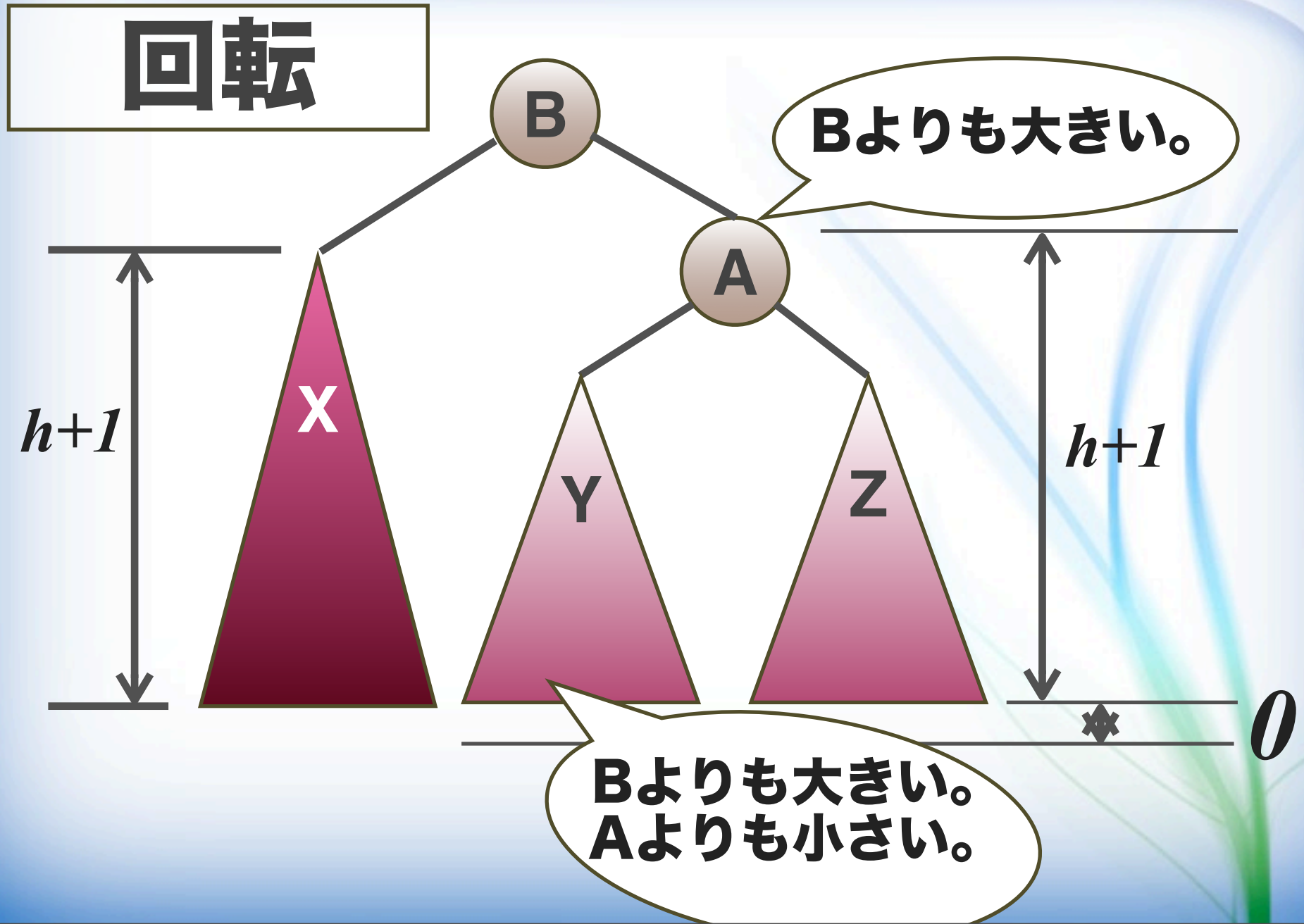
AVL木の整形処理（左側への追加後）



AVL木の整形処理（左側への追加後）

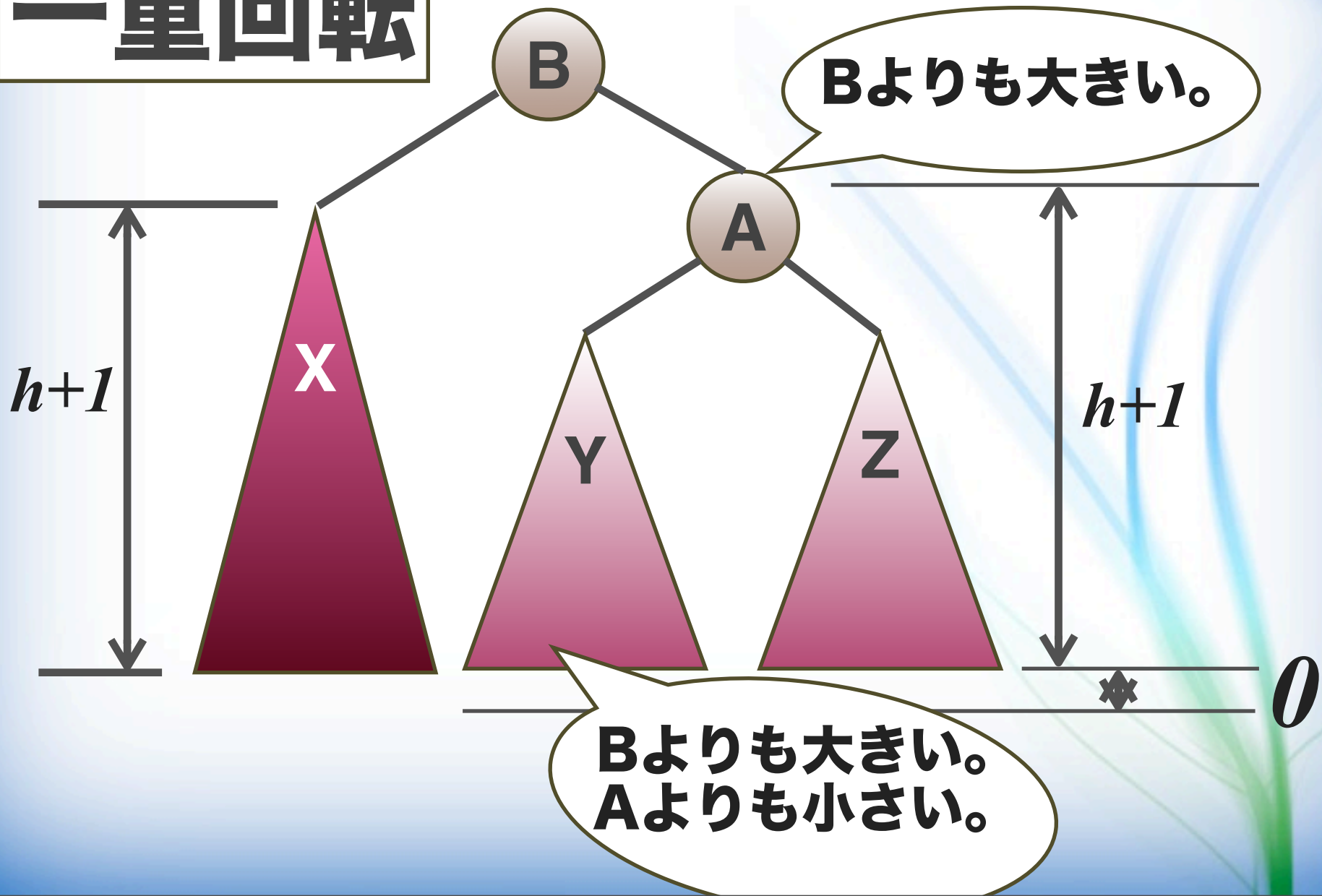


AVL木の整形処理（左側への追加後）

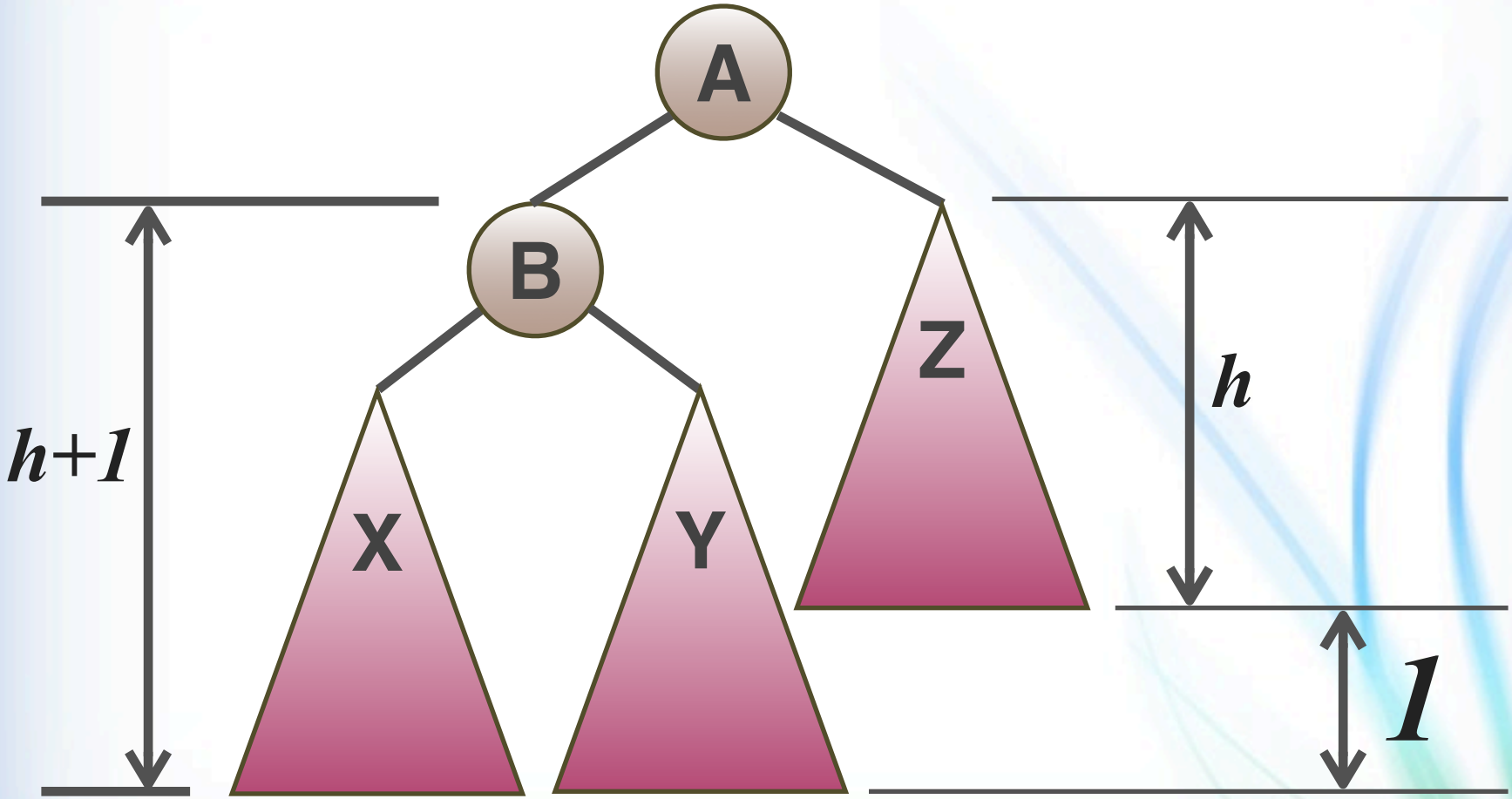


AVL木の整形処理（左側への追加後）

一重回転

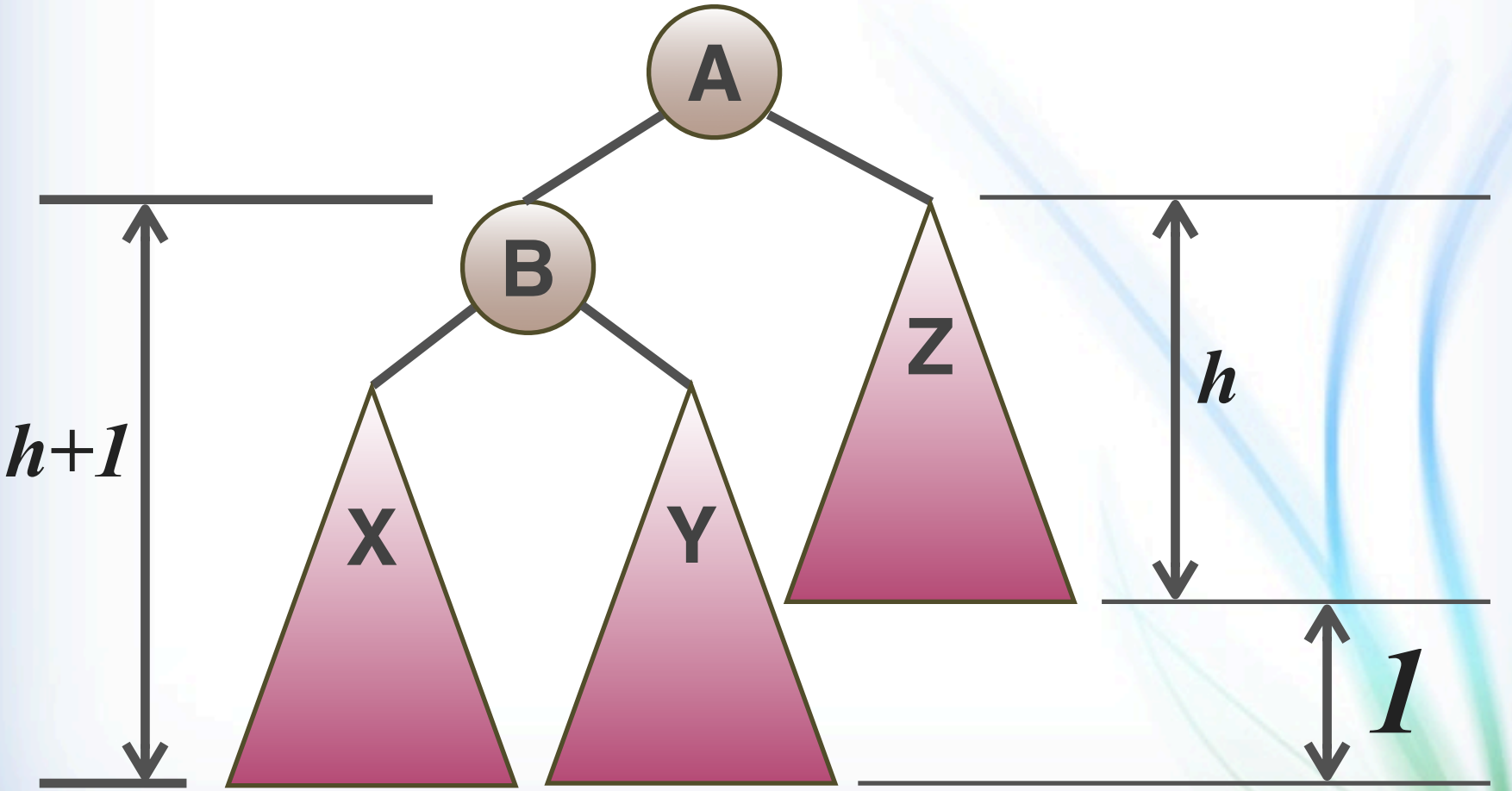


AVL木の基本形（平衡状態）



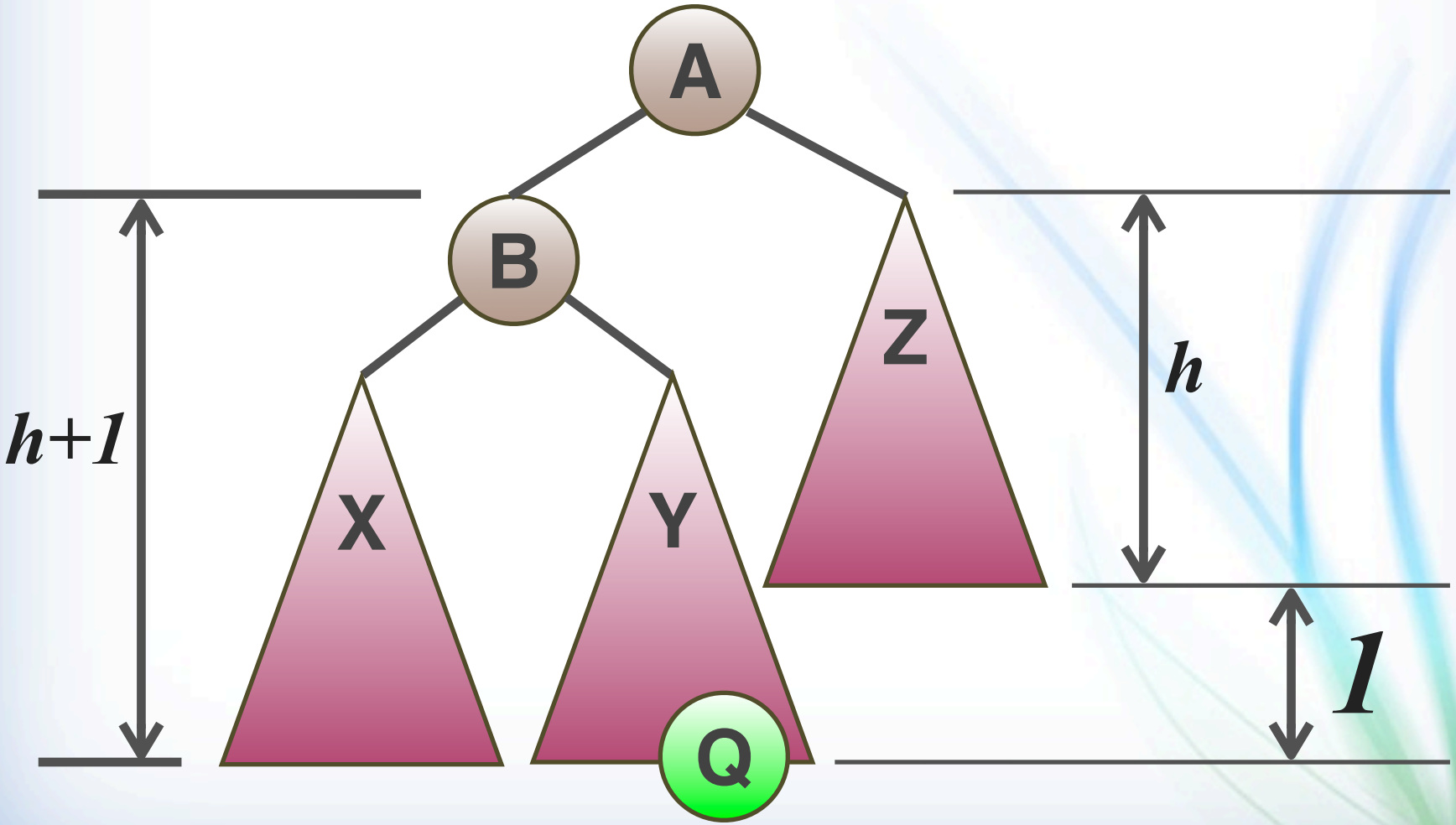
AVL木の追加処理（右側への追加）

Yに要素を追加。



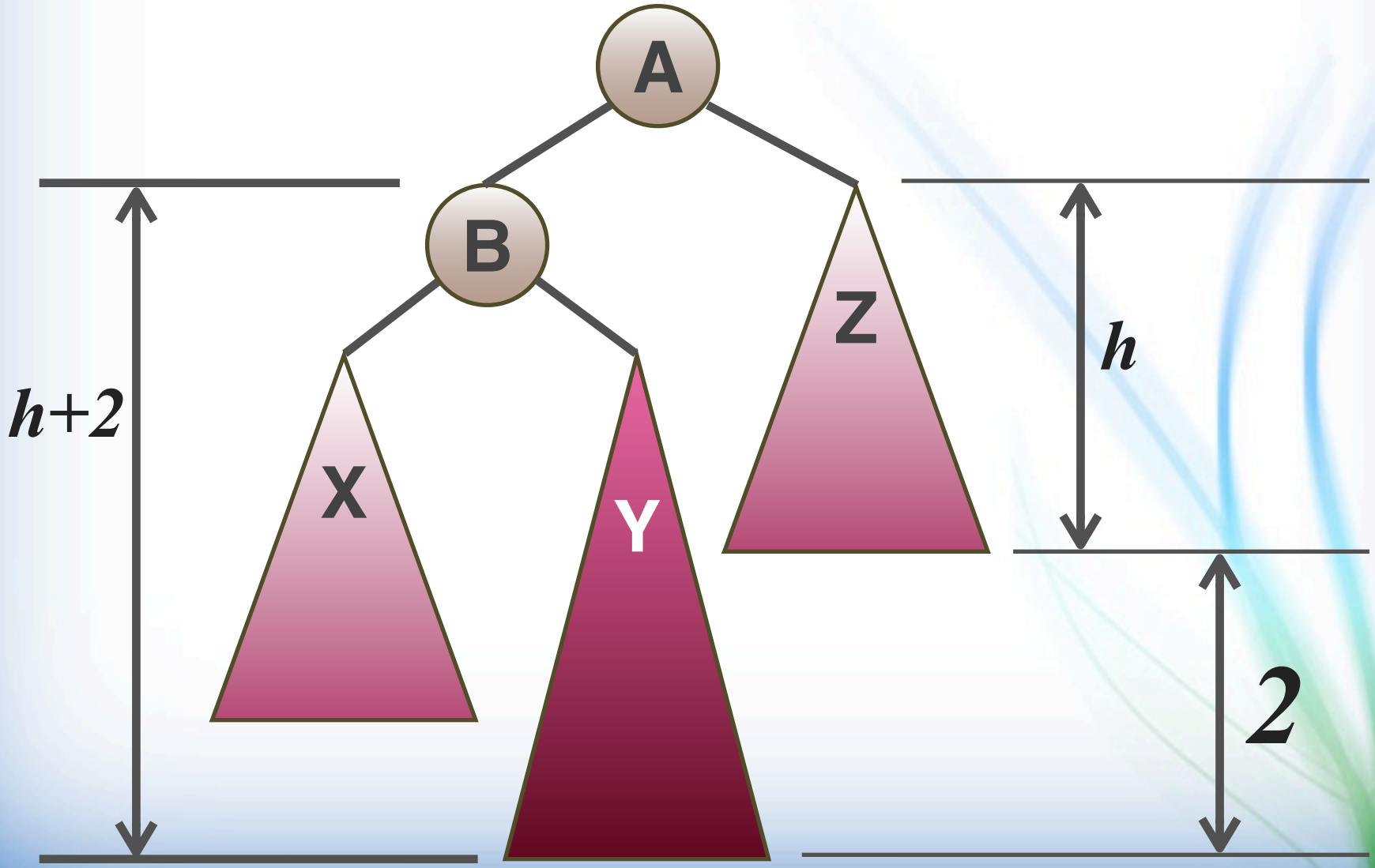
AVL木の追加処理（右側への追加）

Yに要素を追加。



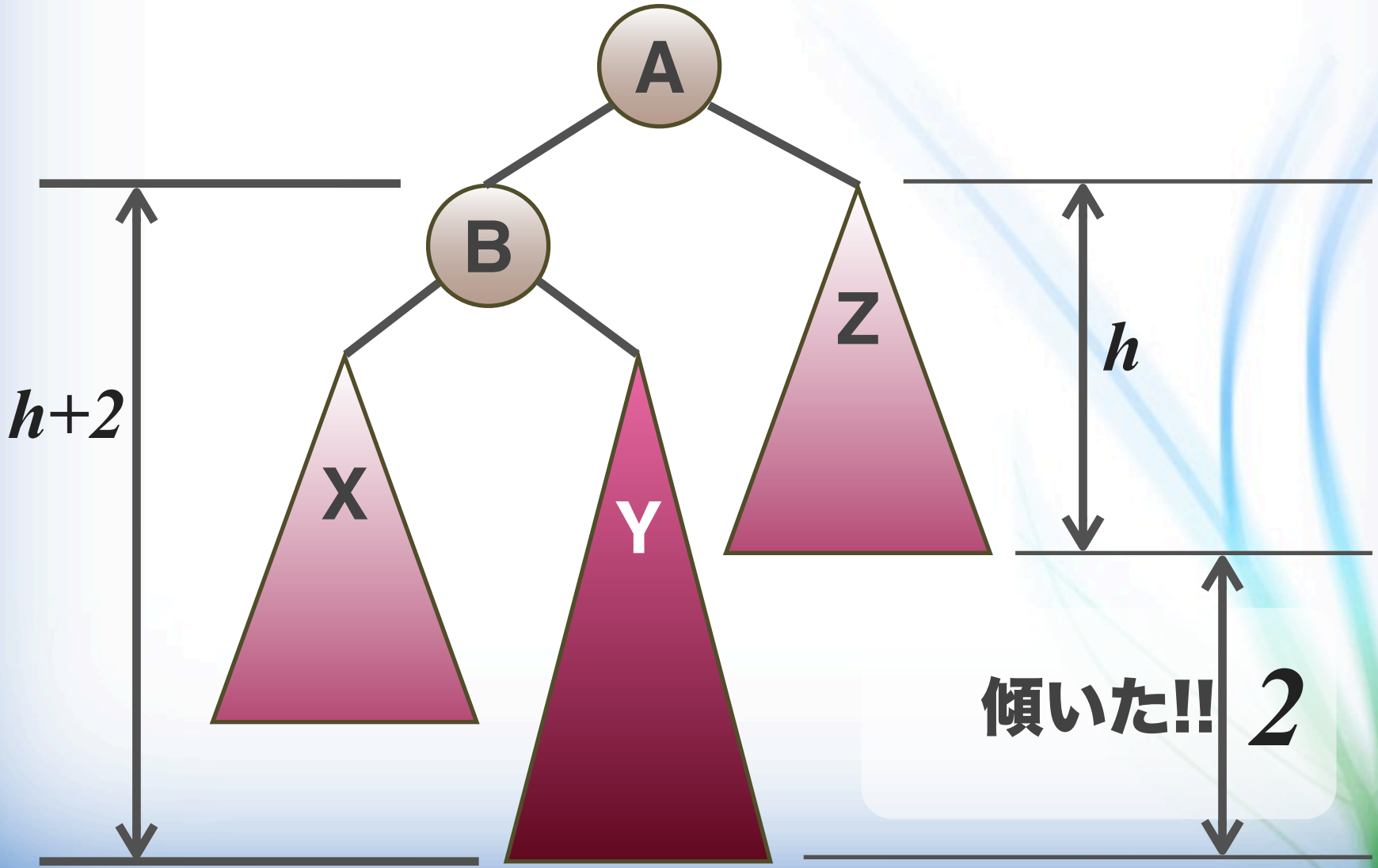
AVL木の追加処理（右側への追加）

Yに要素を追加。



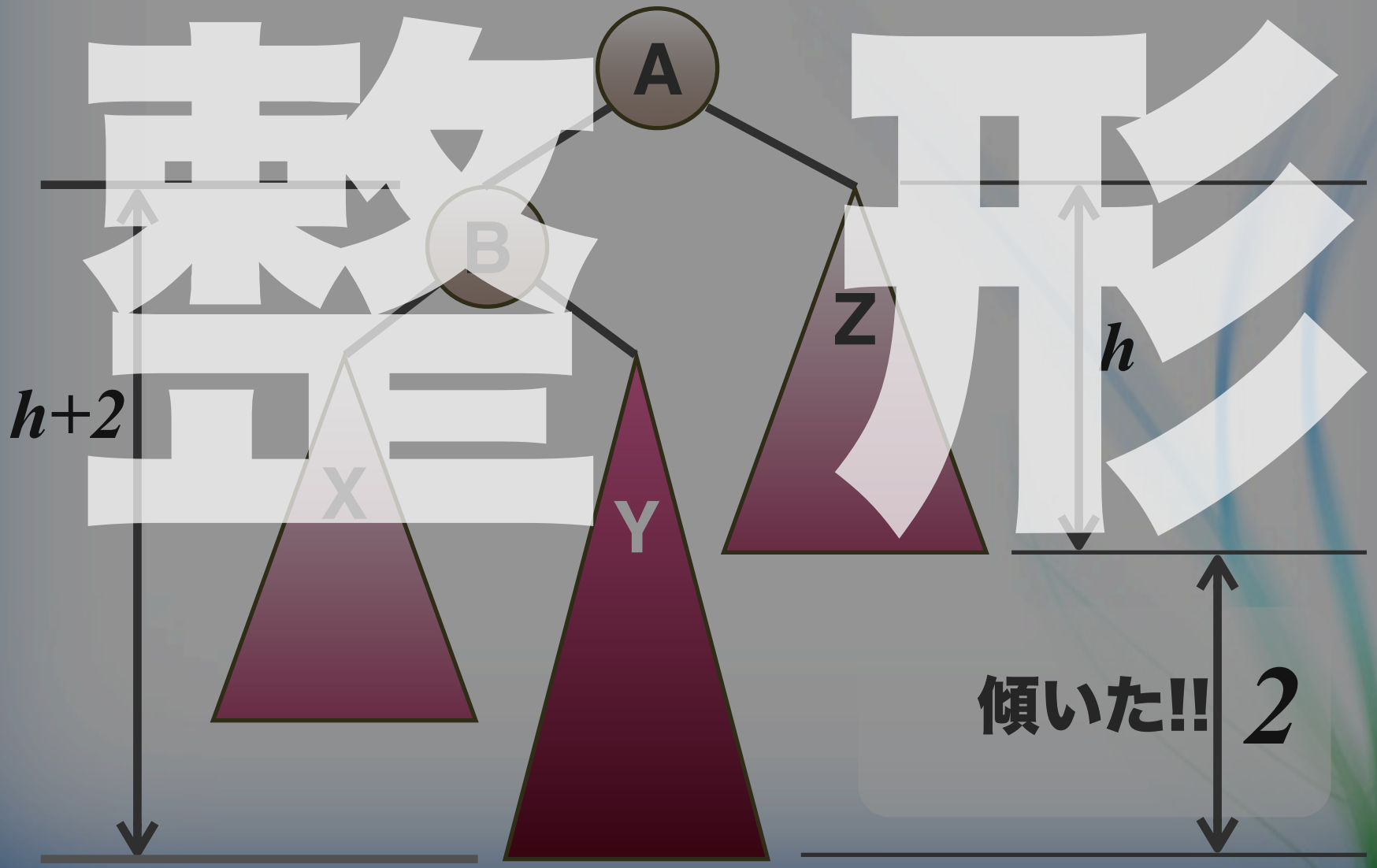
AVL木の追加処理（右側への追加）

Yに要素を追加。



AVL木の追加処理（右側への追加）

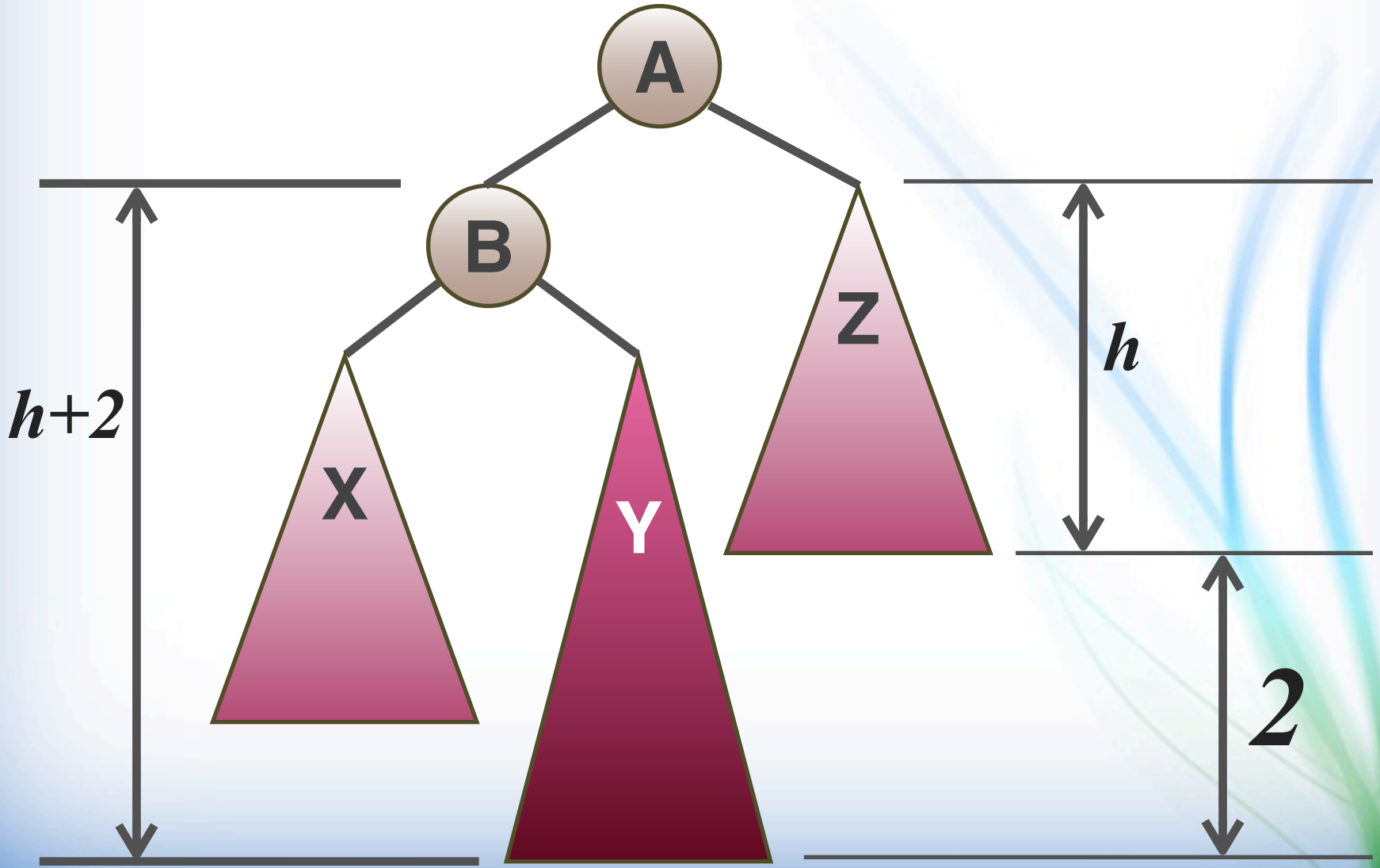
Yに要素を追加。



整形

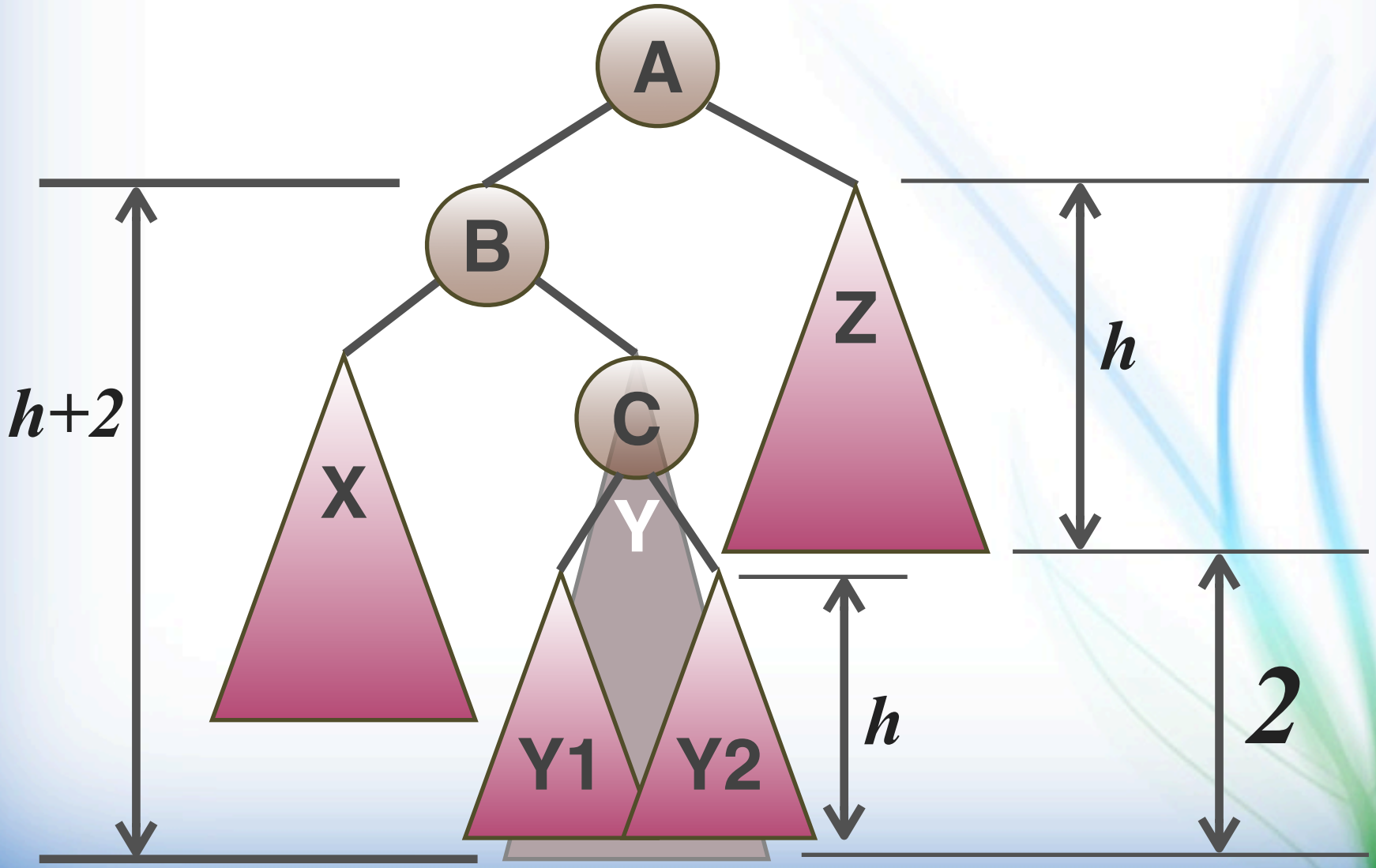
AVL木の追加処理（右側への追加）

Yを分解してみる。

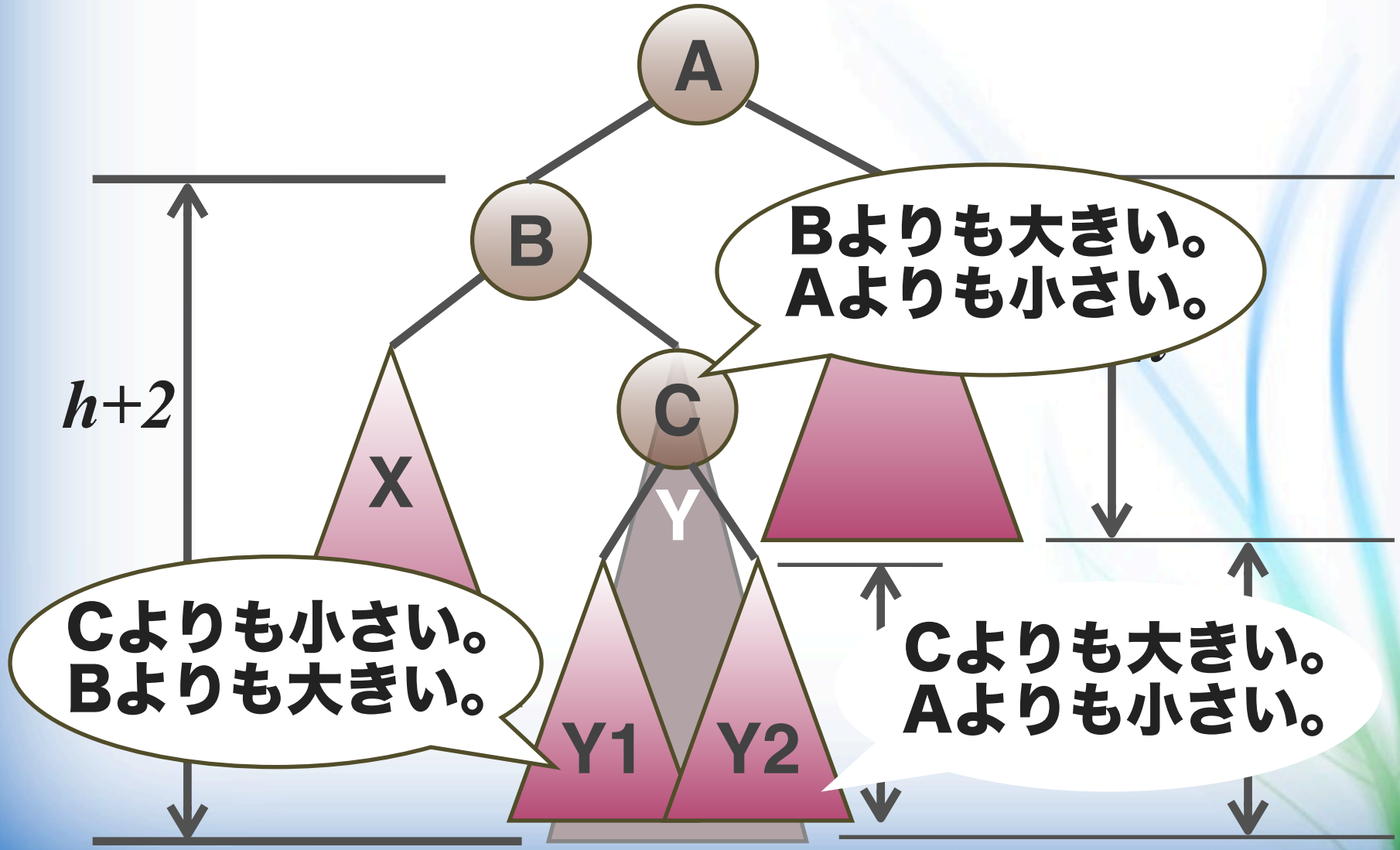


AVL木の追加処理（右側への追加）

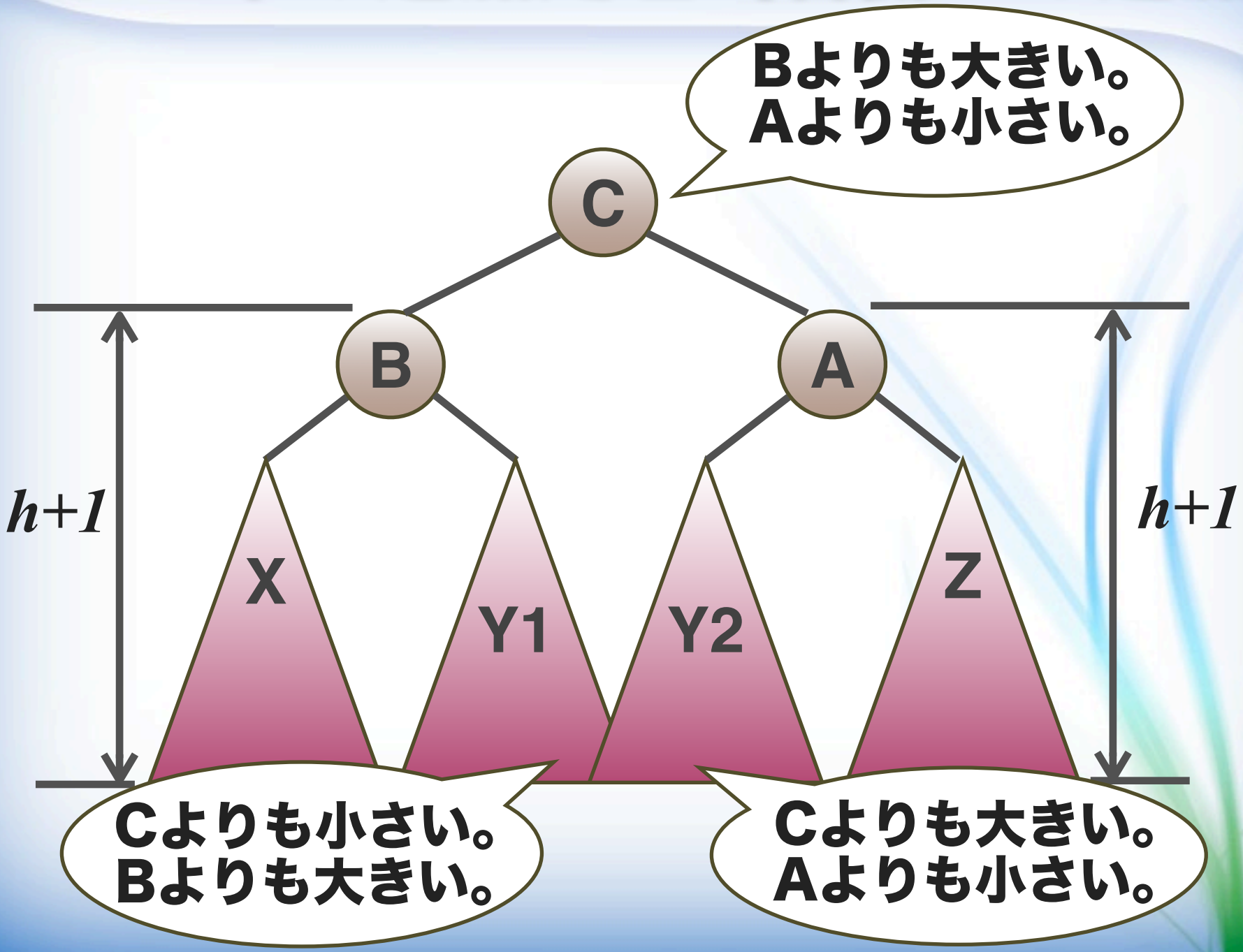
Yを分解してみる。



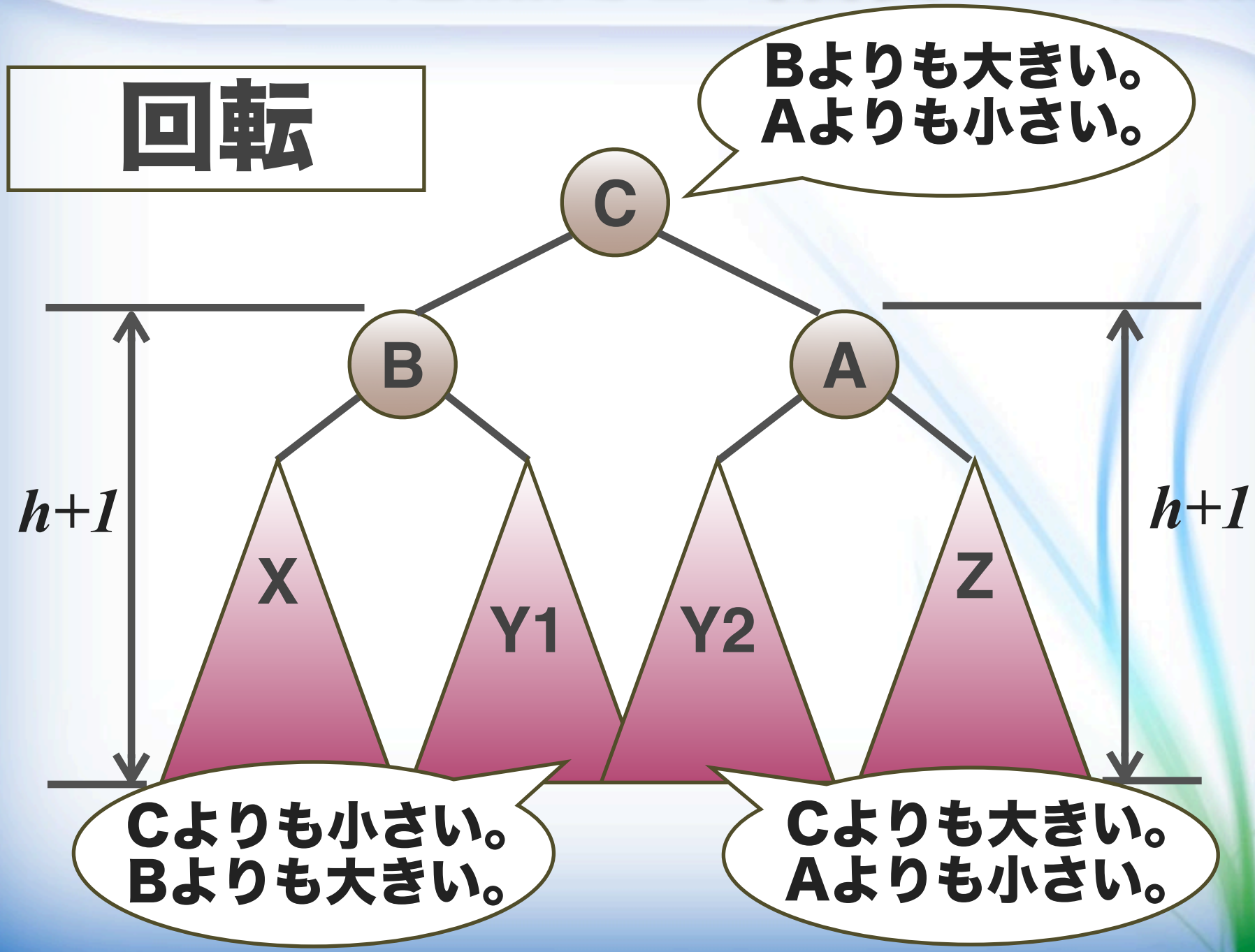
AVL木の追加処理（右側への追加）



AVL木の追加処理（右側への追加）

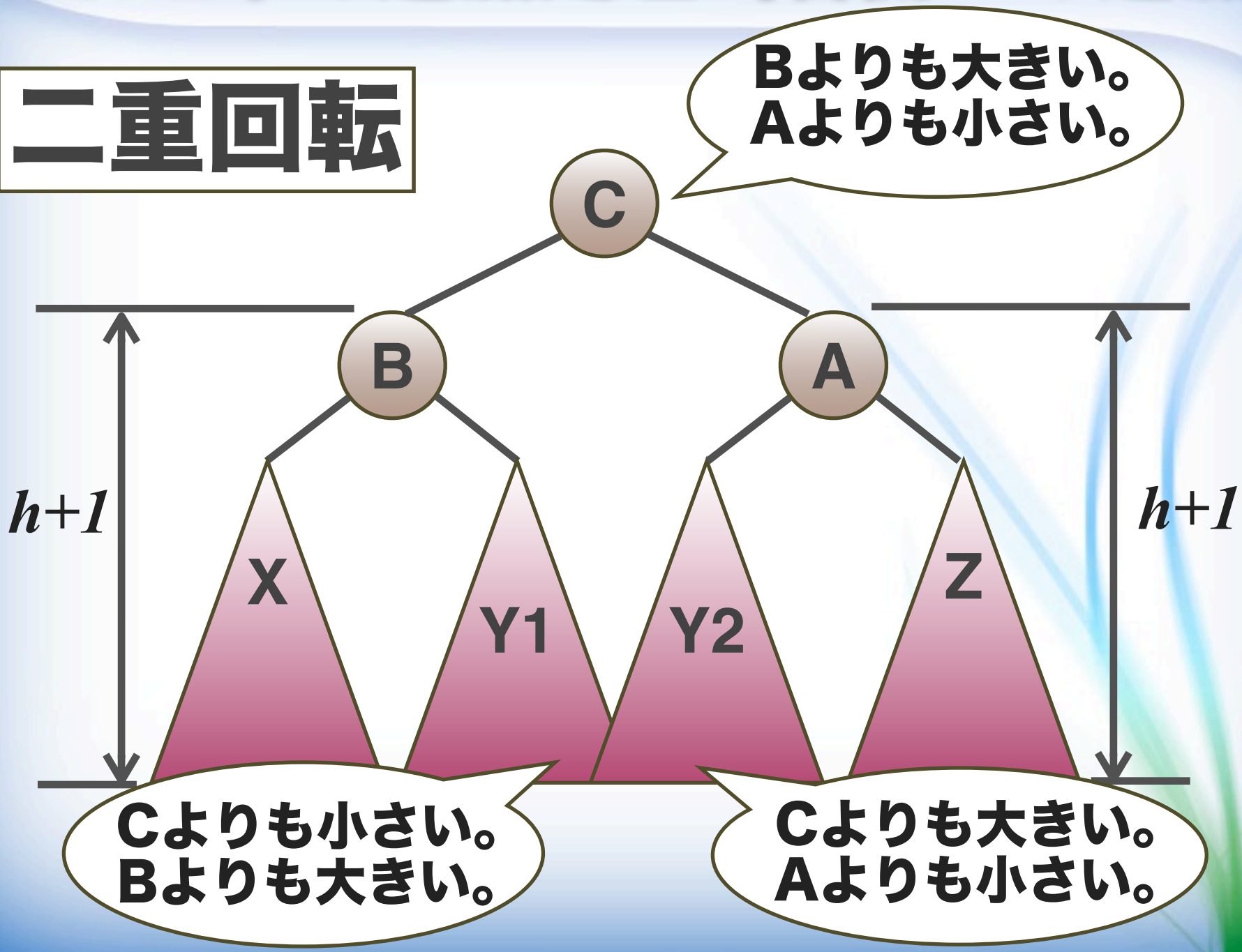


AVL木の追加処理（右側への追加）



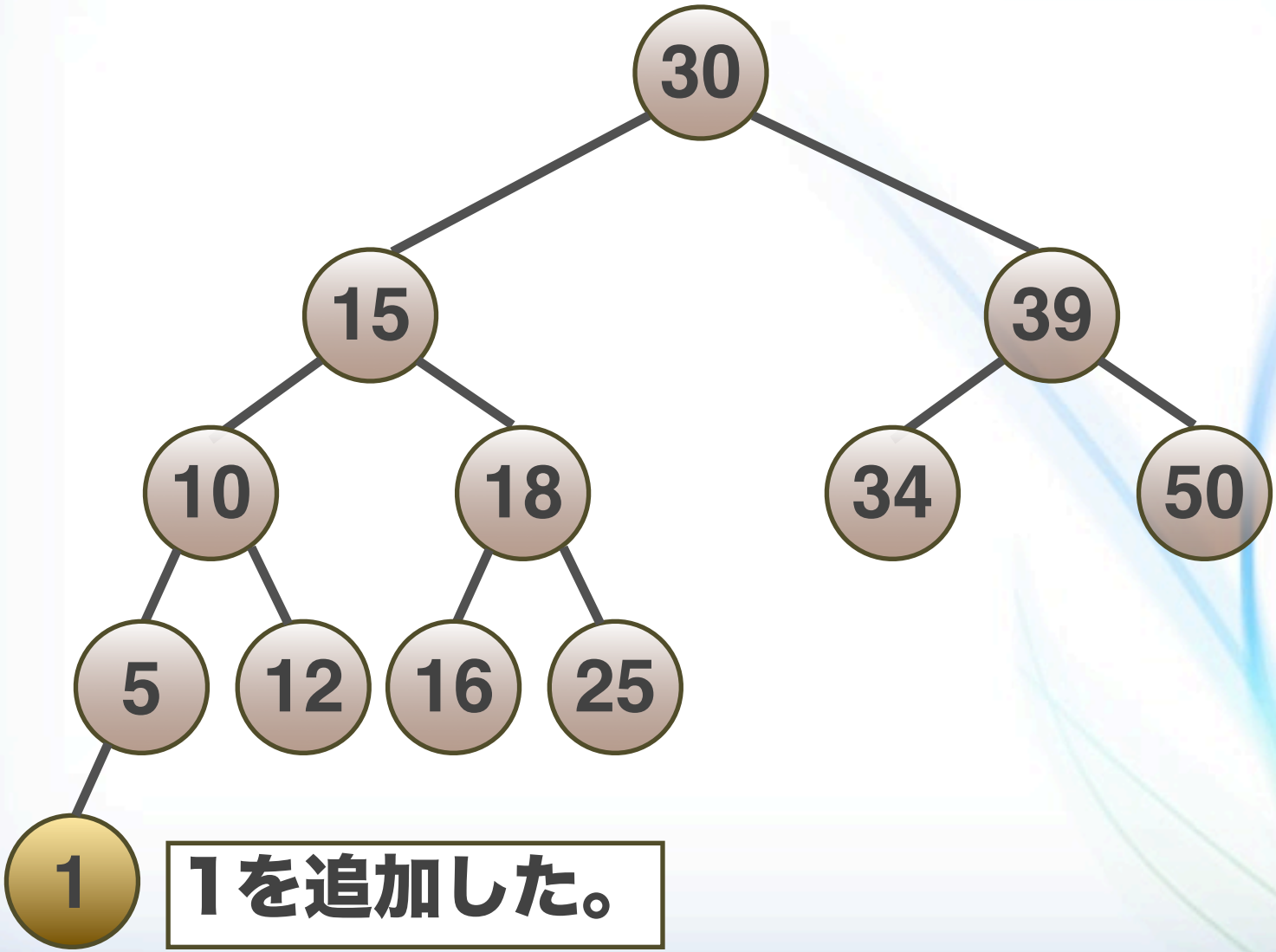
AVL木の追加処理（右側への追加）

二重回転

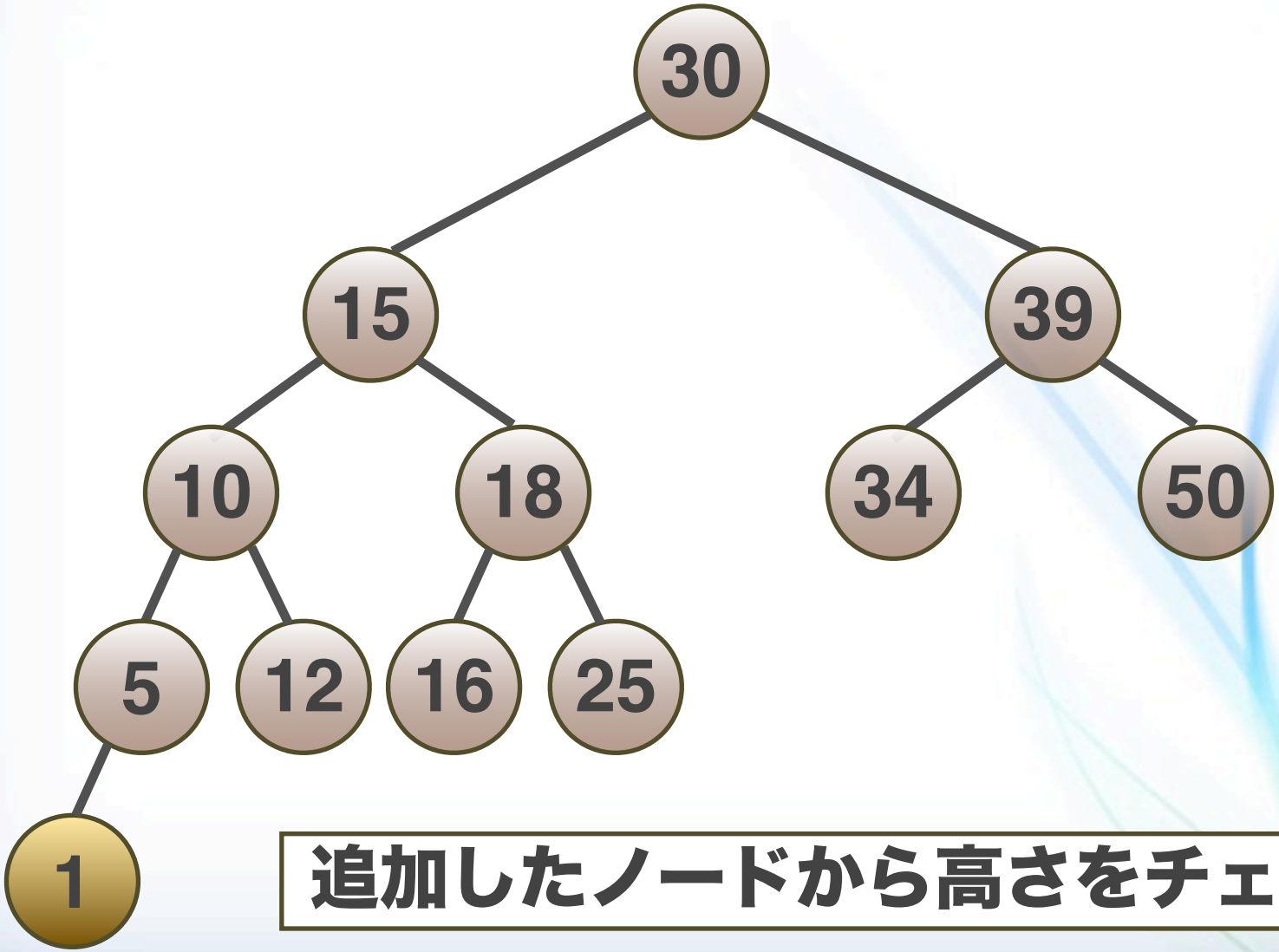


AVL木の実例

AVL木の実例１

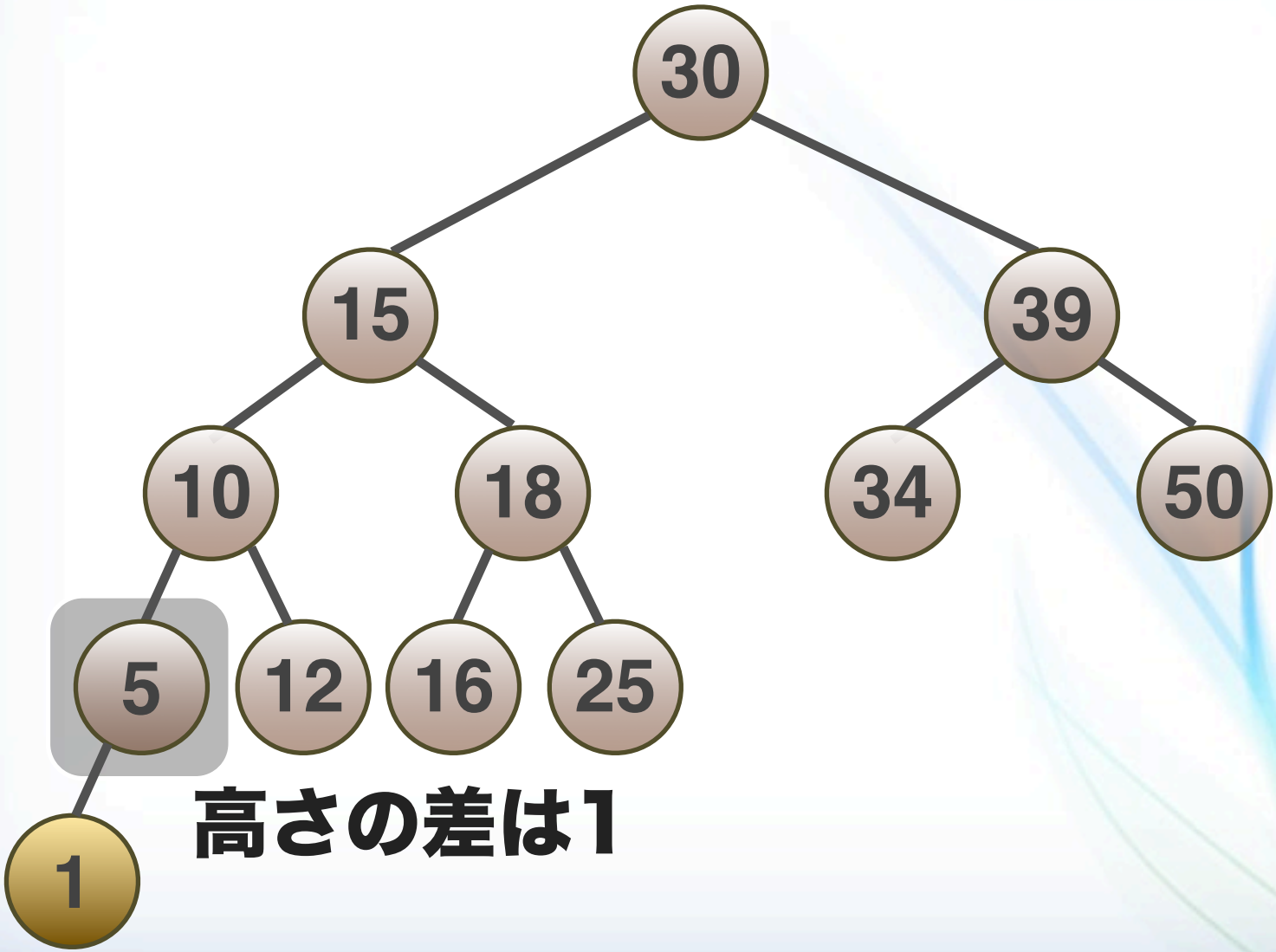


AVL木の実例１

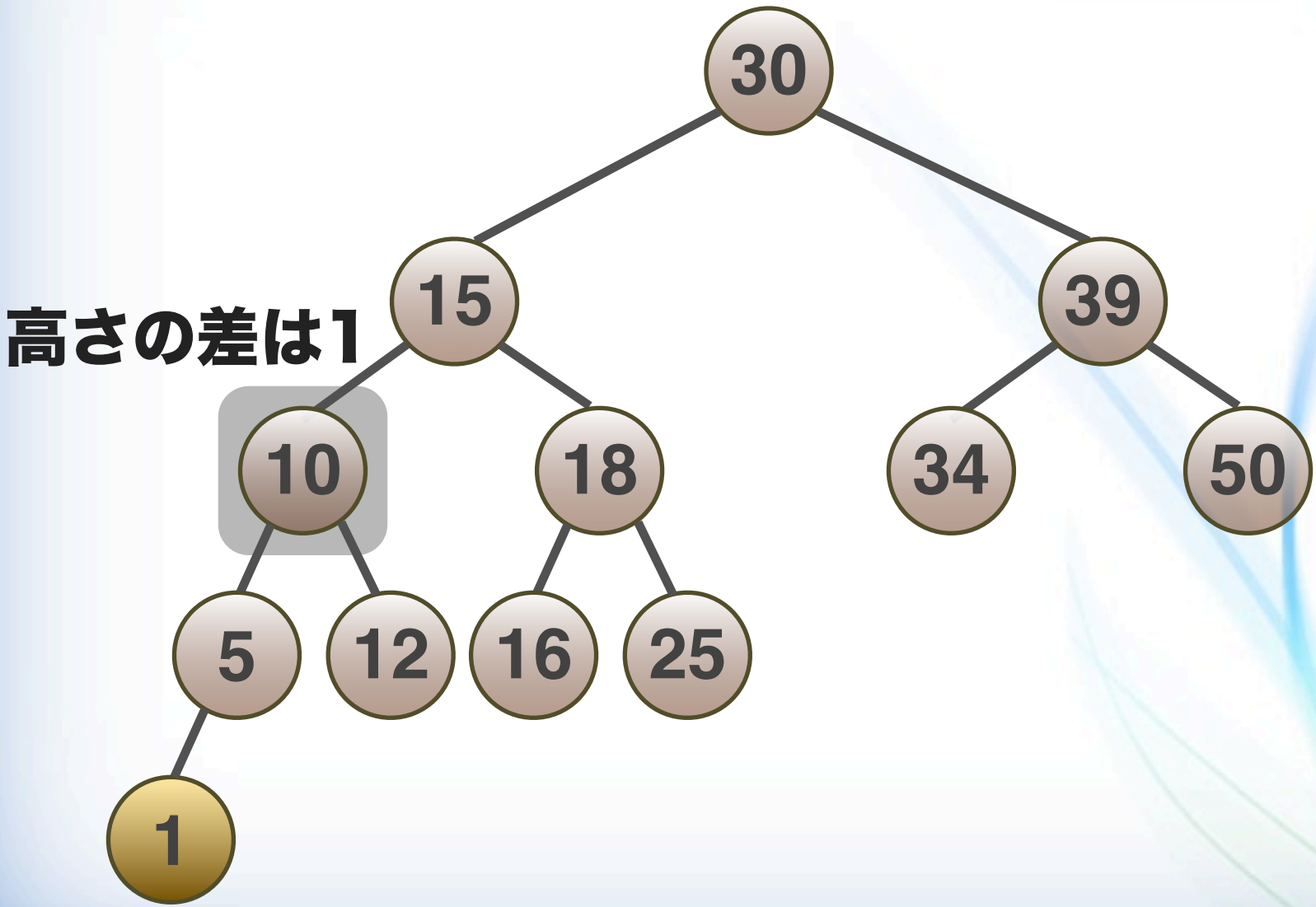


追加したノードから高さをチェック。

AVL木の実例１

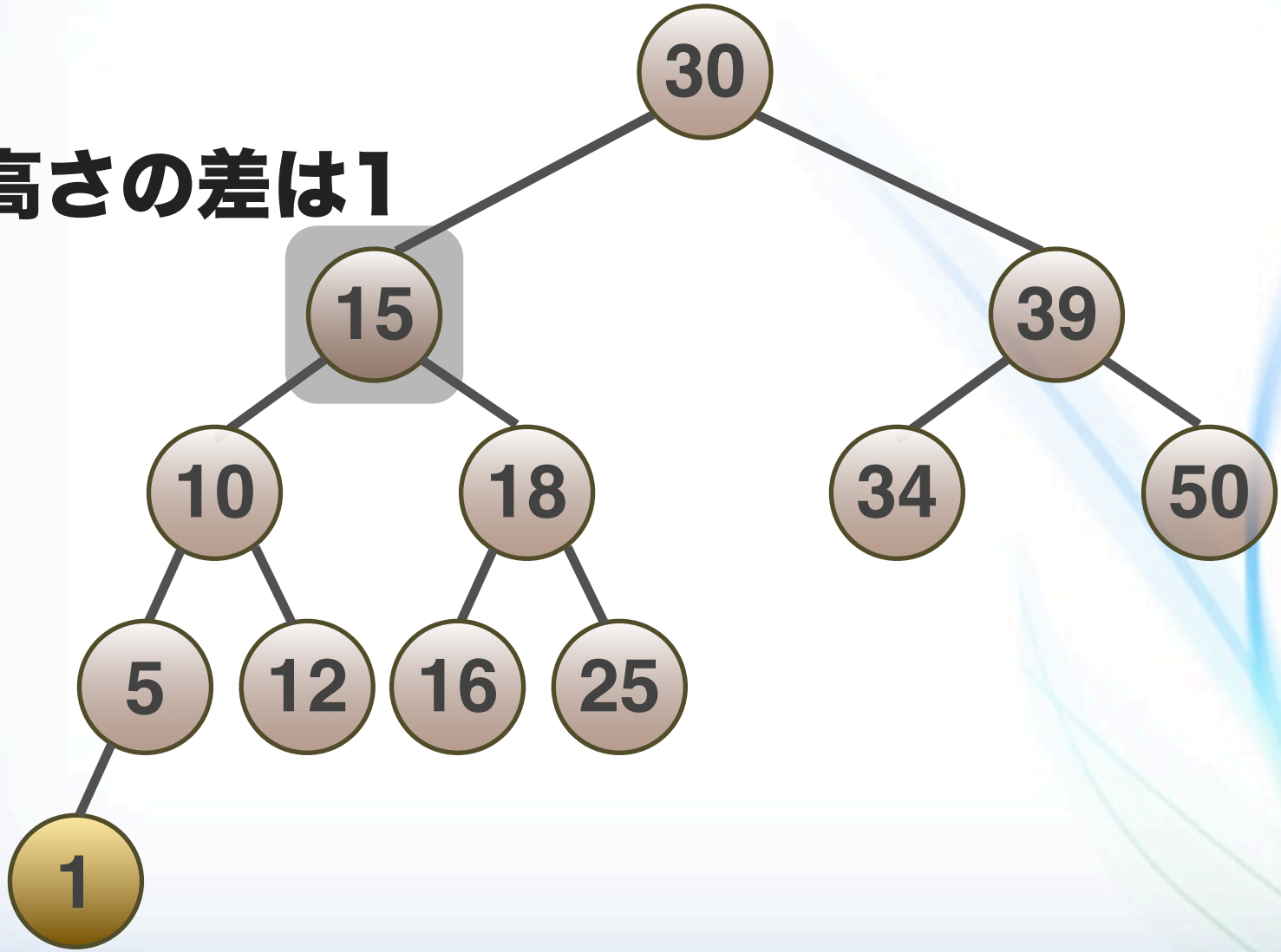


AVL木の実例１



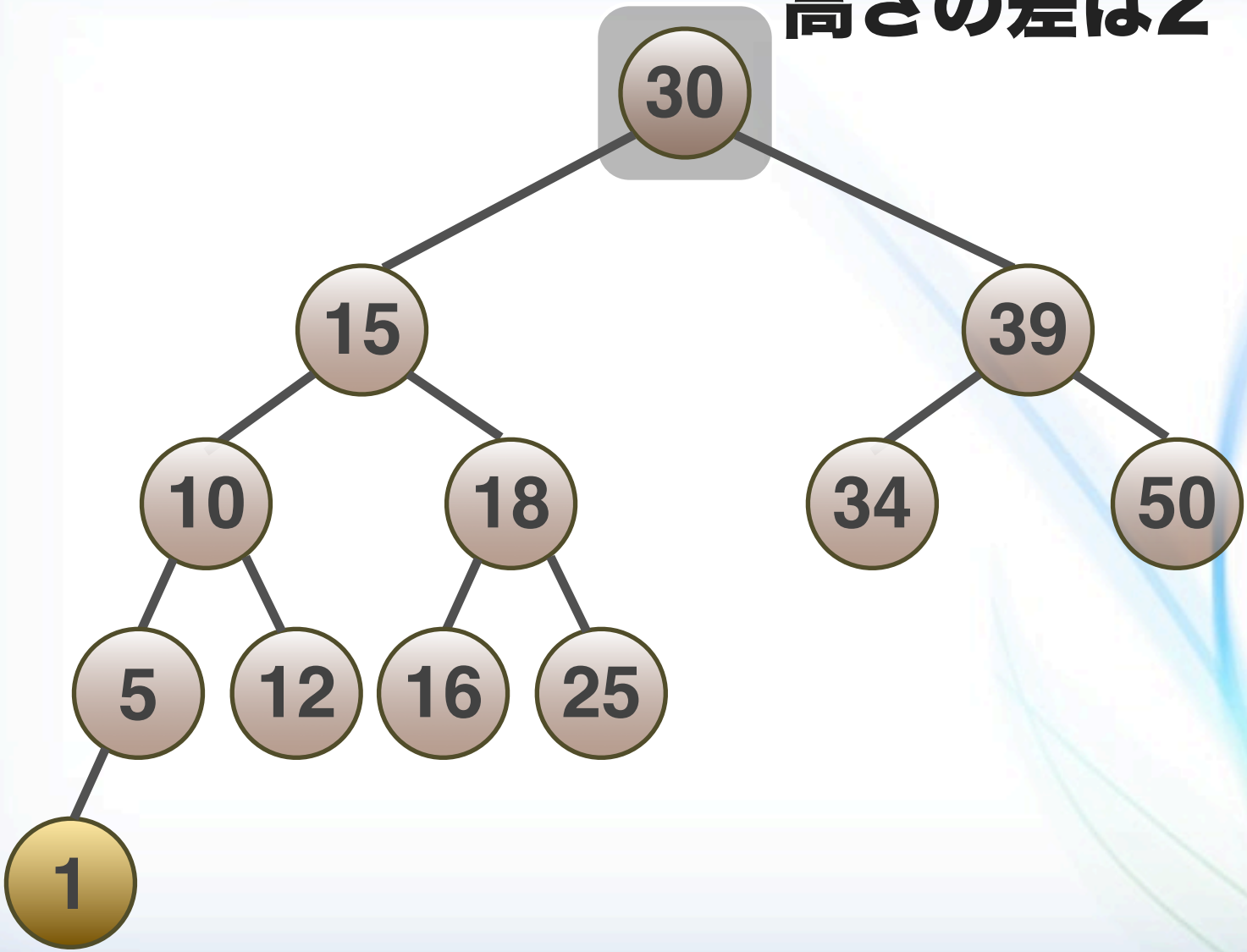
AVL木の実例１

高さの差は1



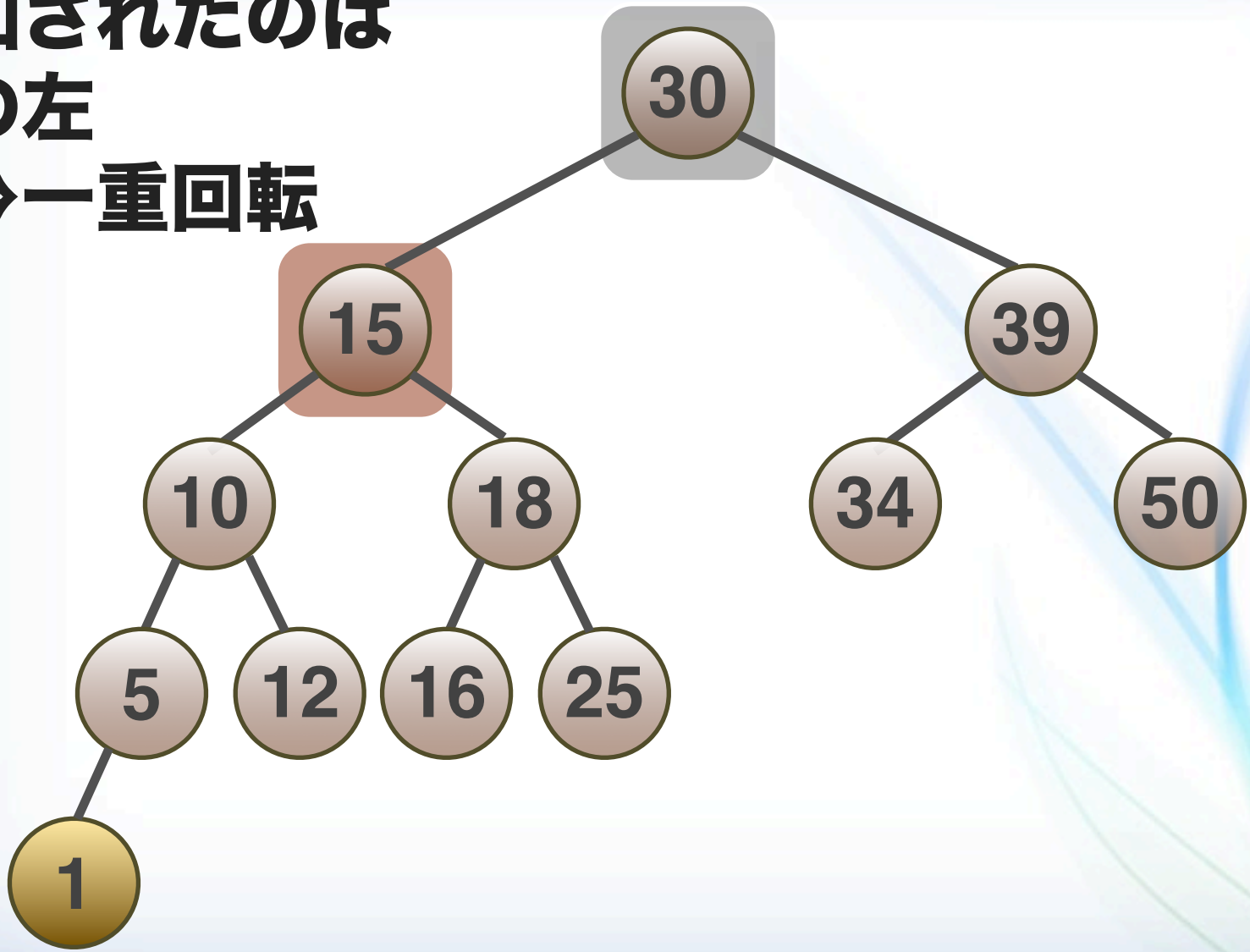
AVL木の実例１

高さの差は2



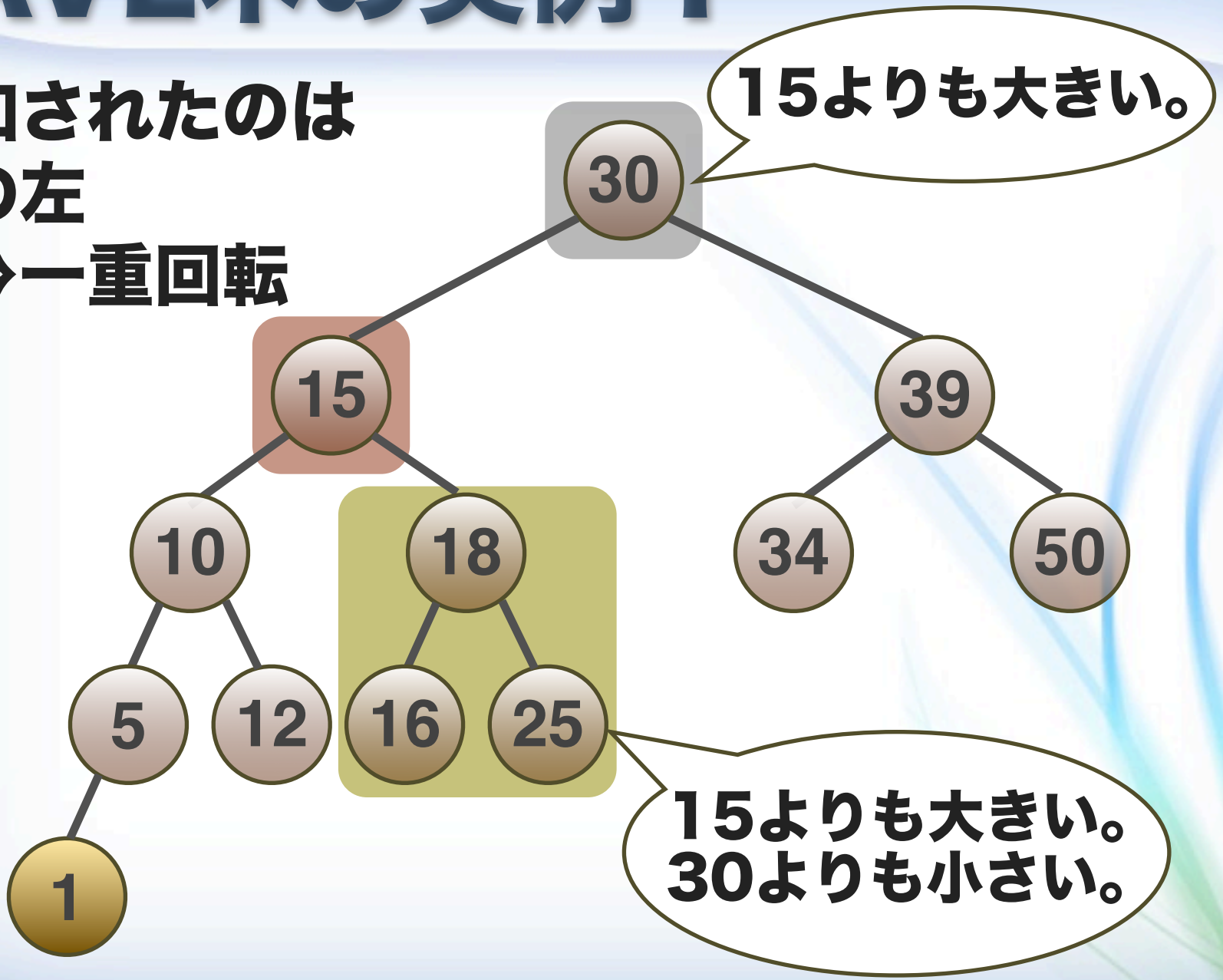
AVL木の実例１

追加されたのは
左の左
⇒一重回転



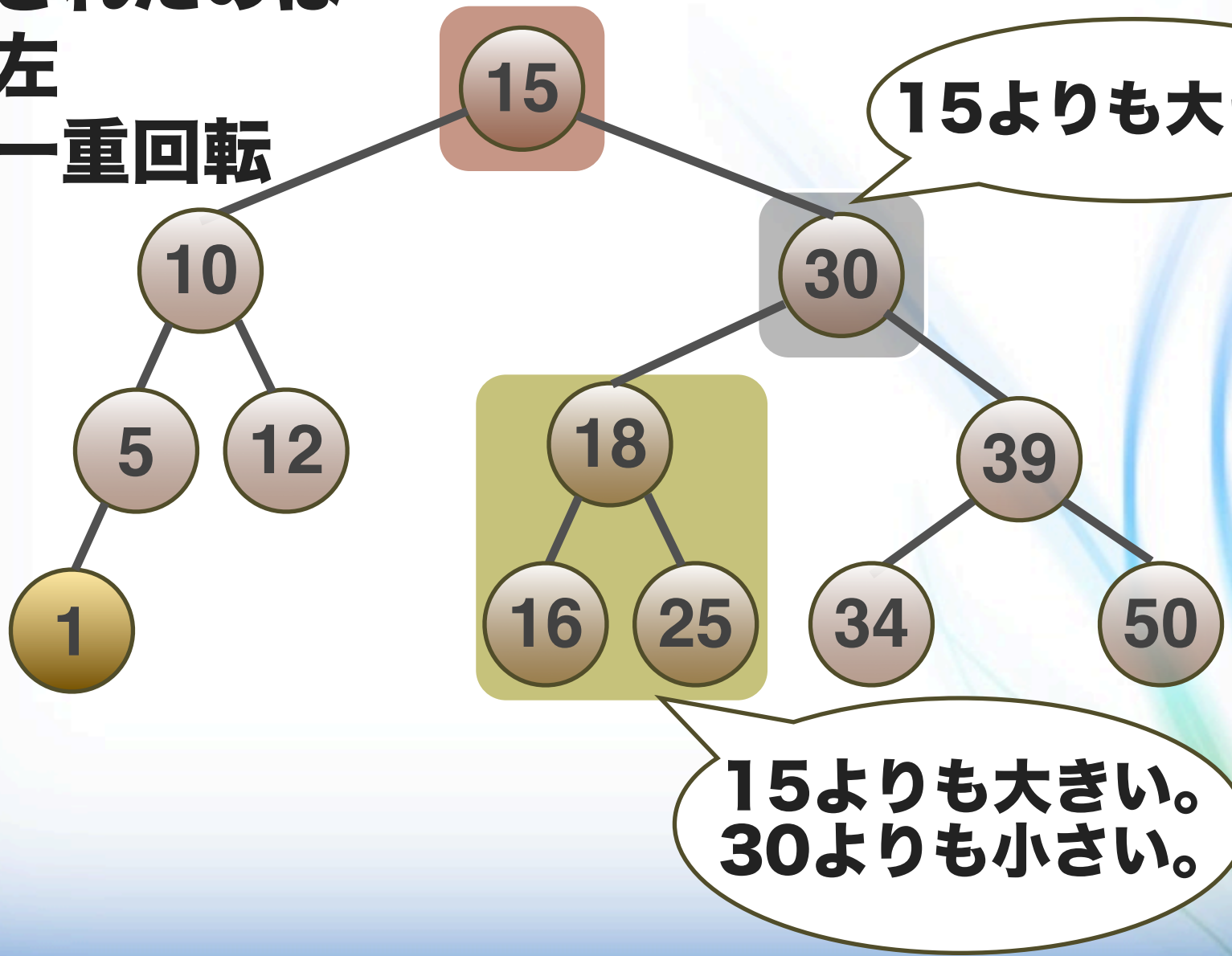
AVL木の実例１

追加されたのは
左の左
⇒一重回転

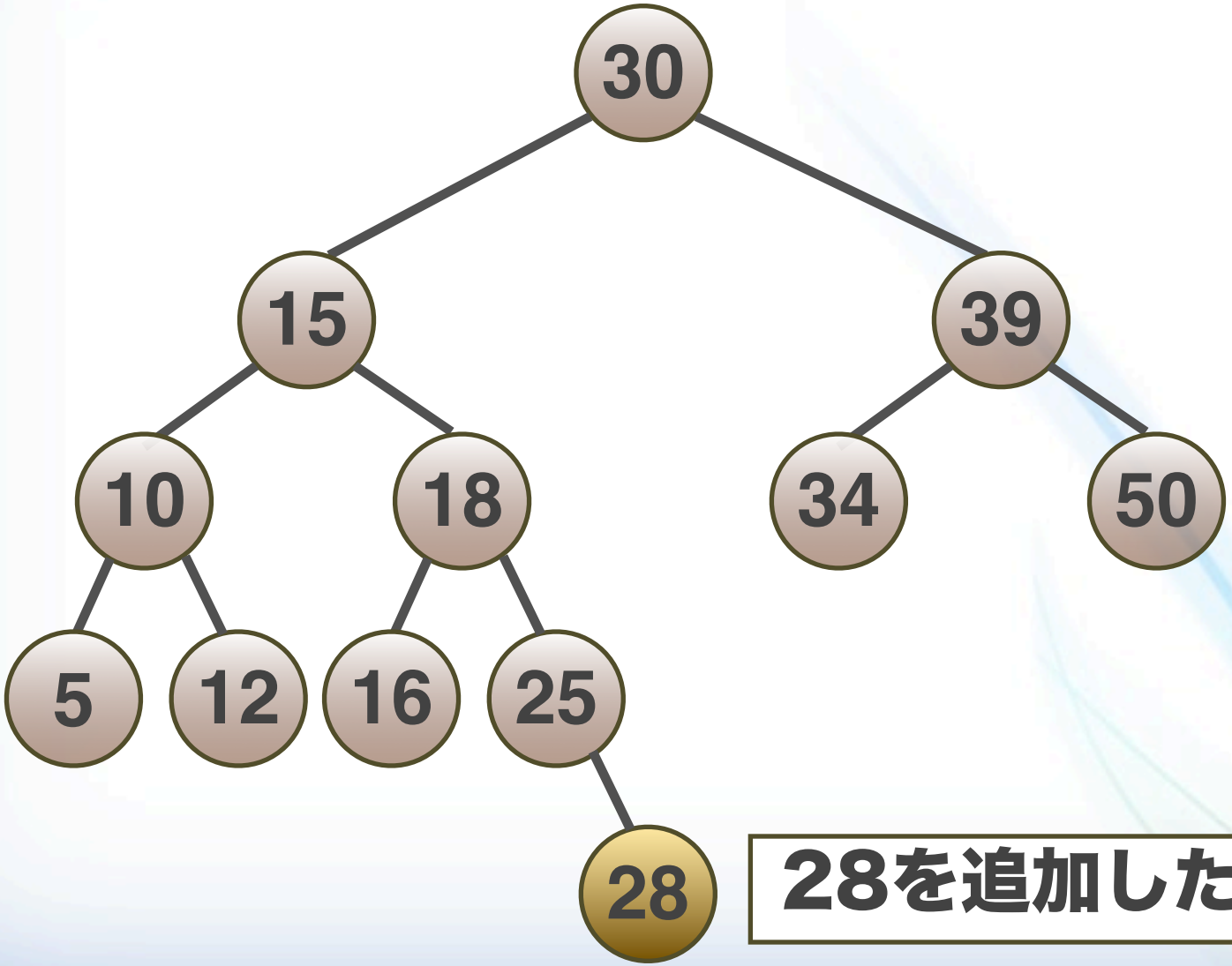


AVL木の実例１

追加されたのは
左の左
⇒一重回転

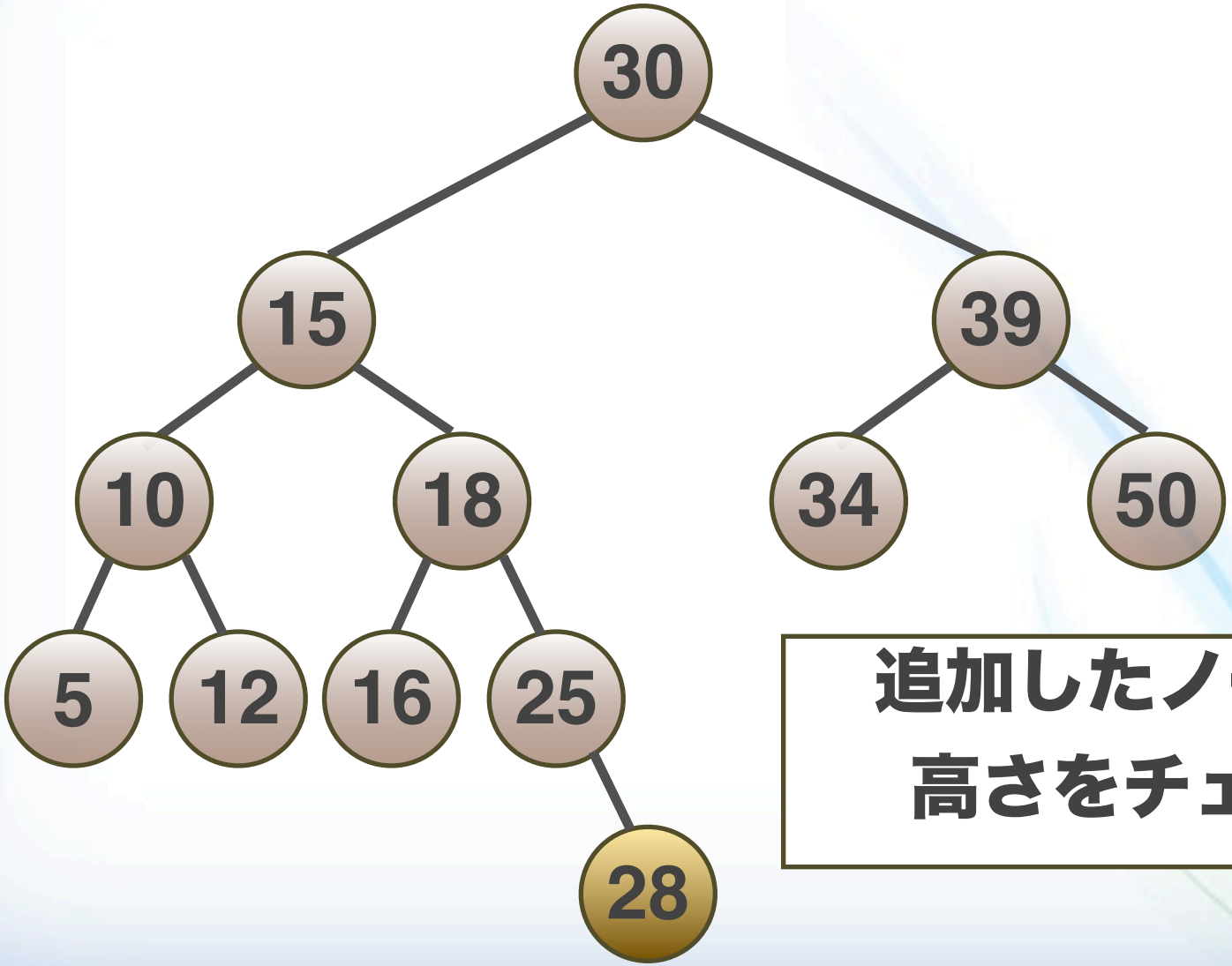


AVL木の実例２



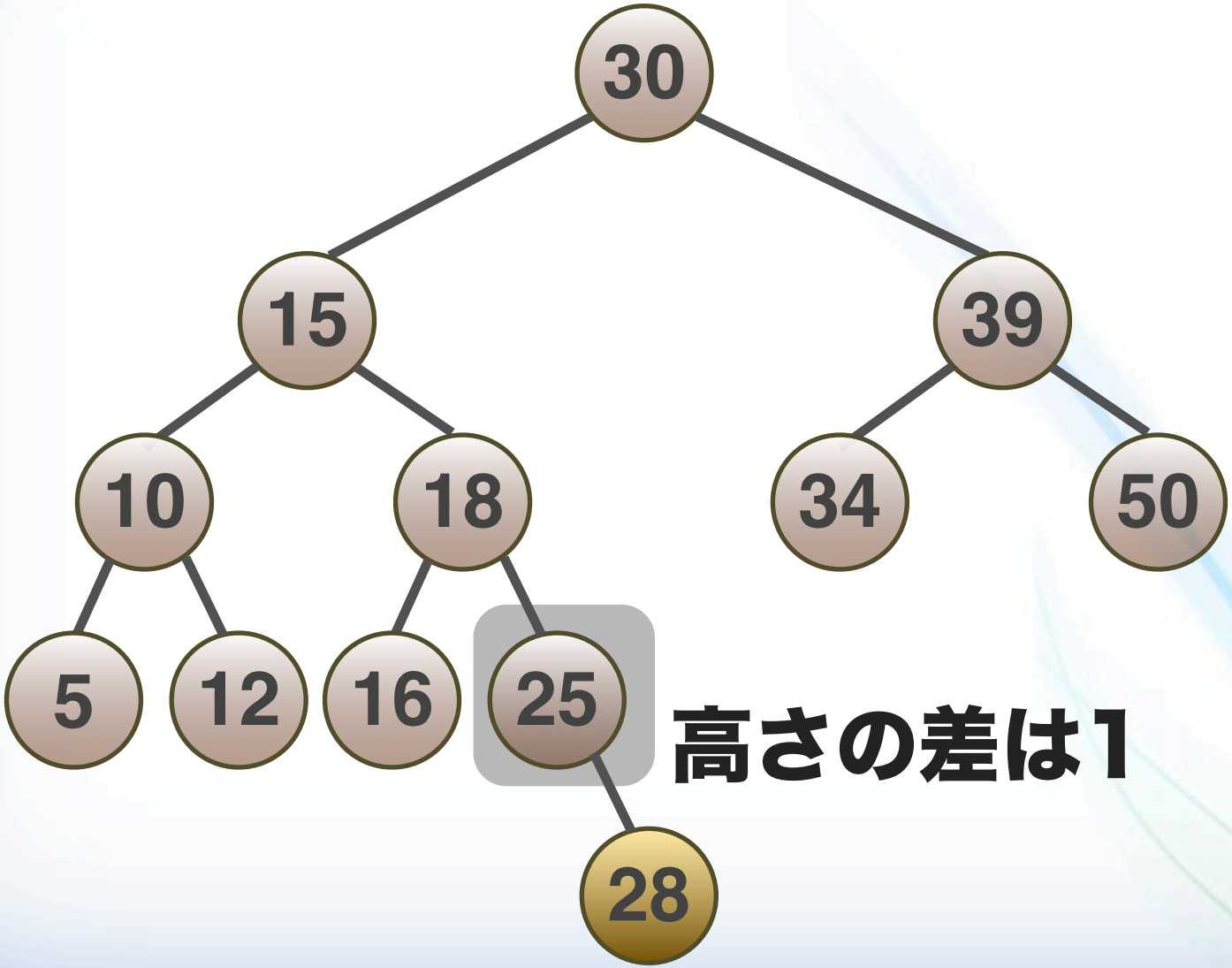
28を追加した。

AVL木の実例２

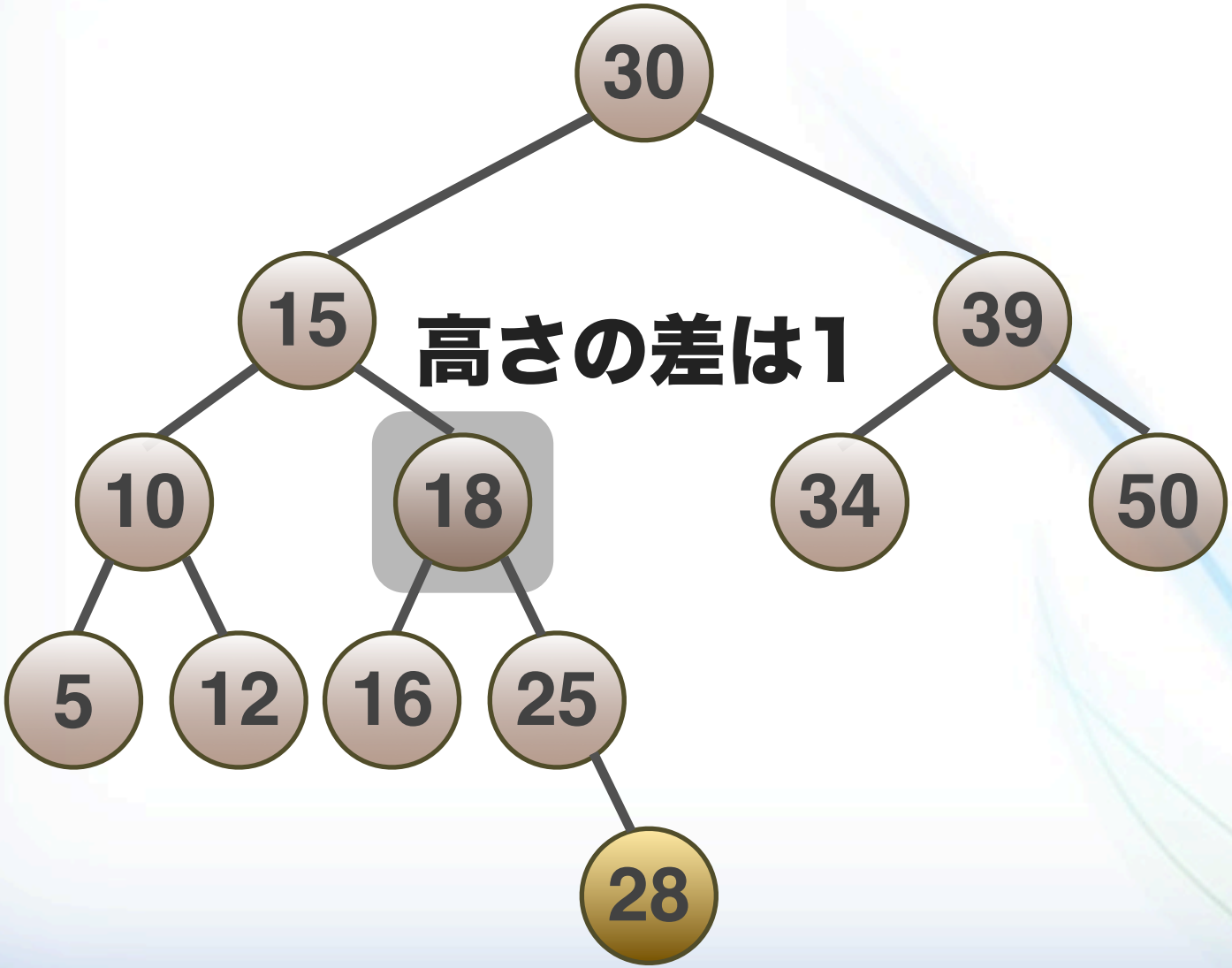


追加したノードから
高さをチェック。

AVL木の実例２

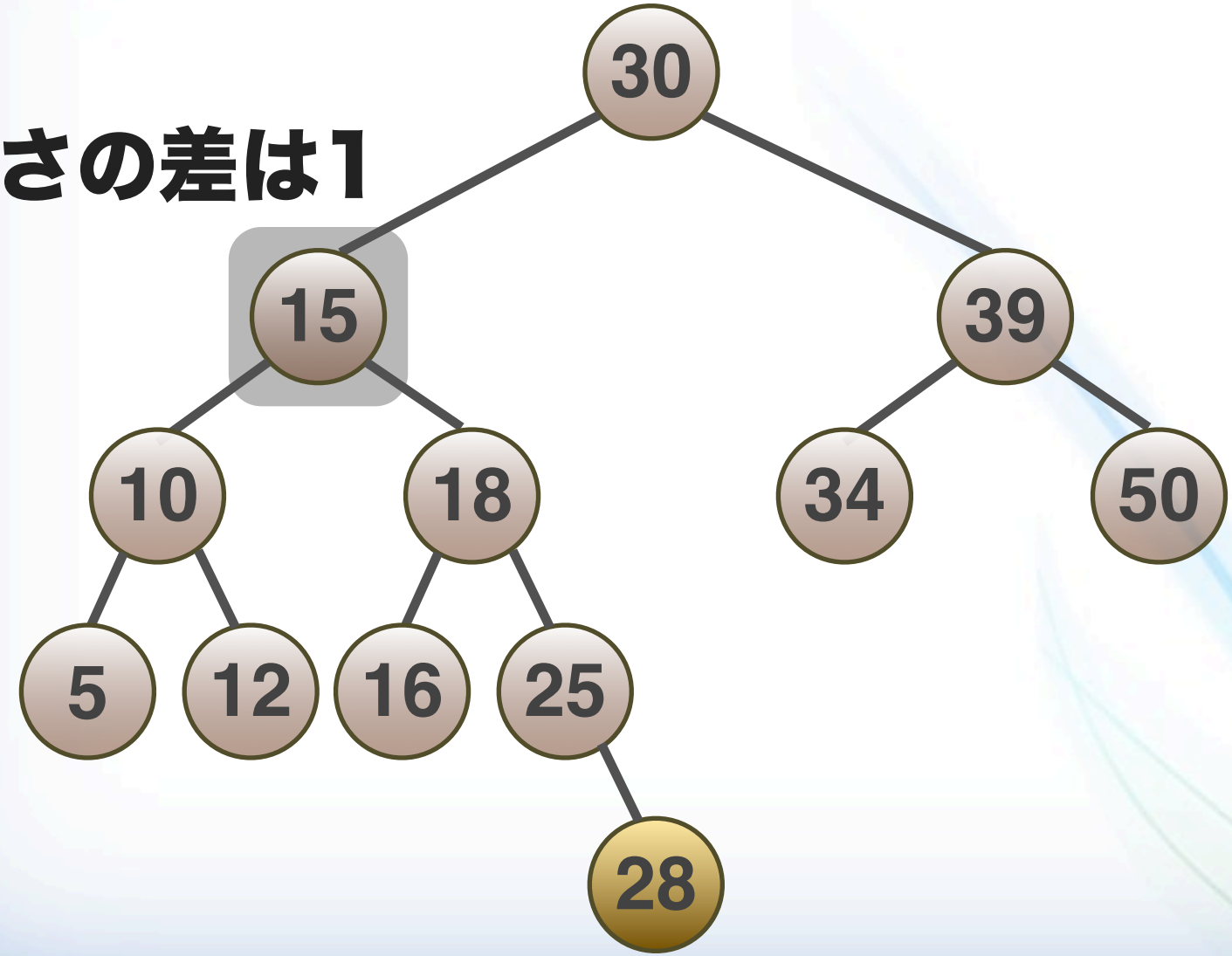


AVL木の実例２



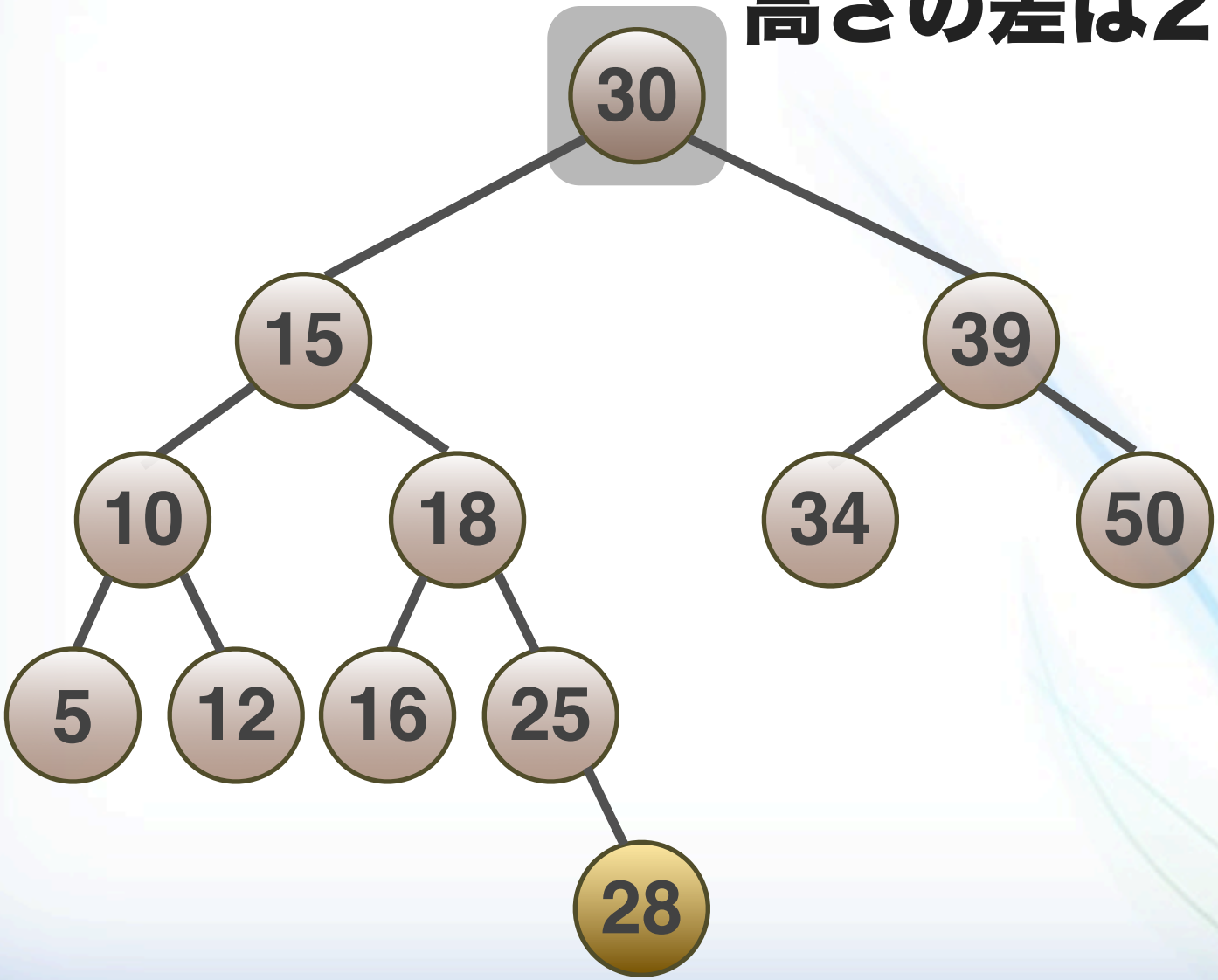
AVL木の実例２

高さの差は1



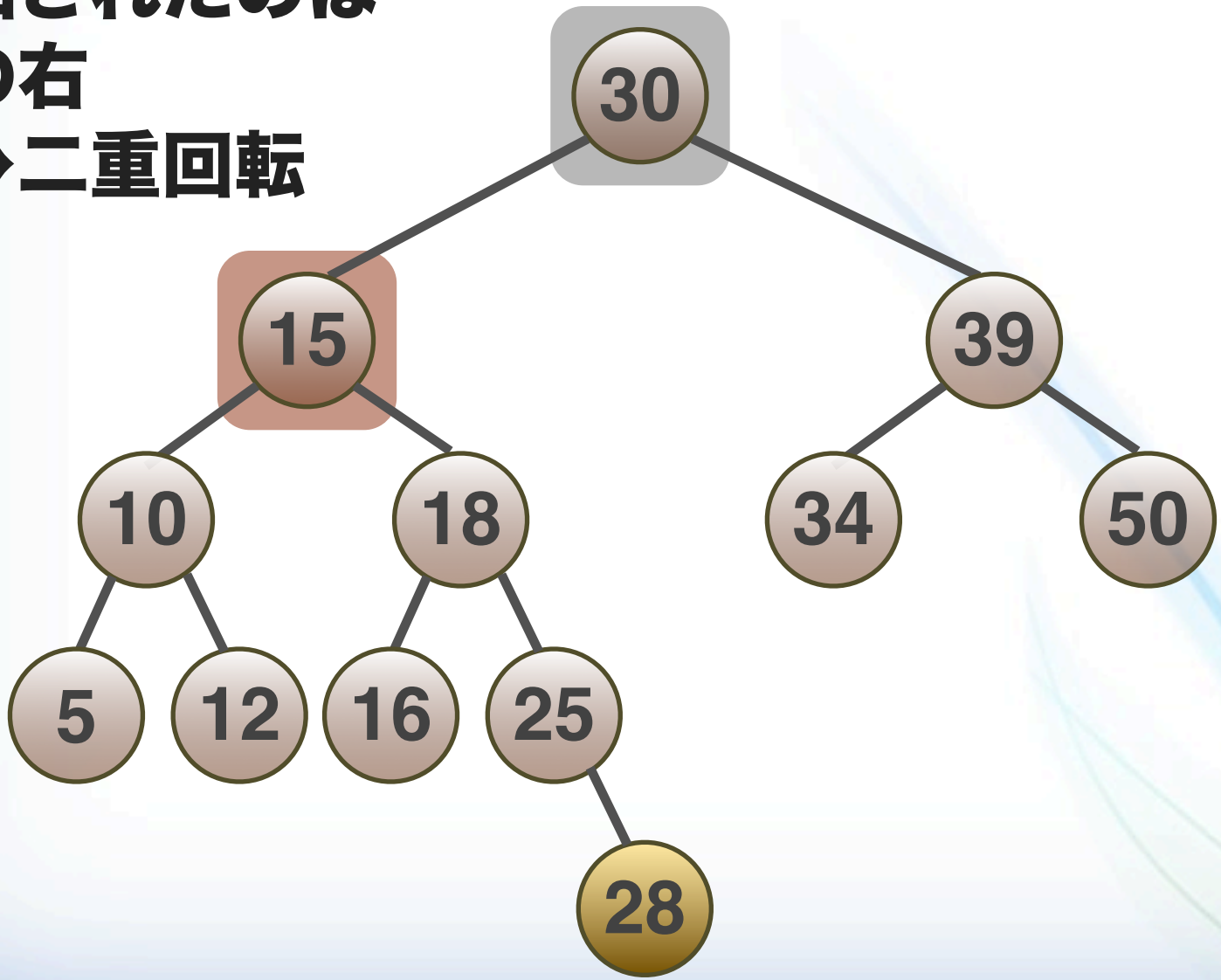
AVL木の実例２

高さの差は2



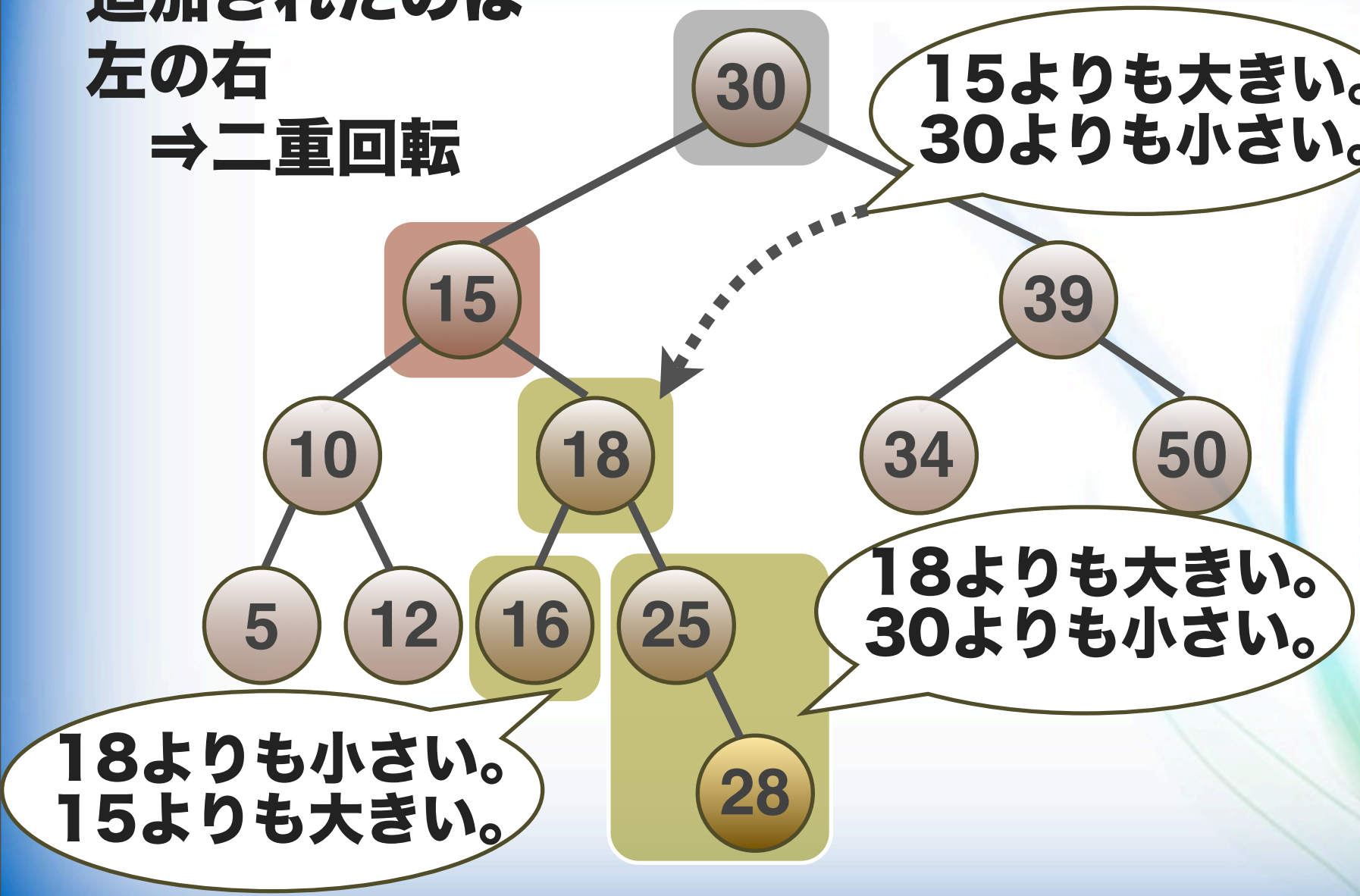
AVL木の実例２

追加されたのは
左の右
⇒二重回転



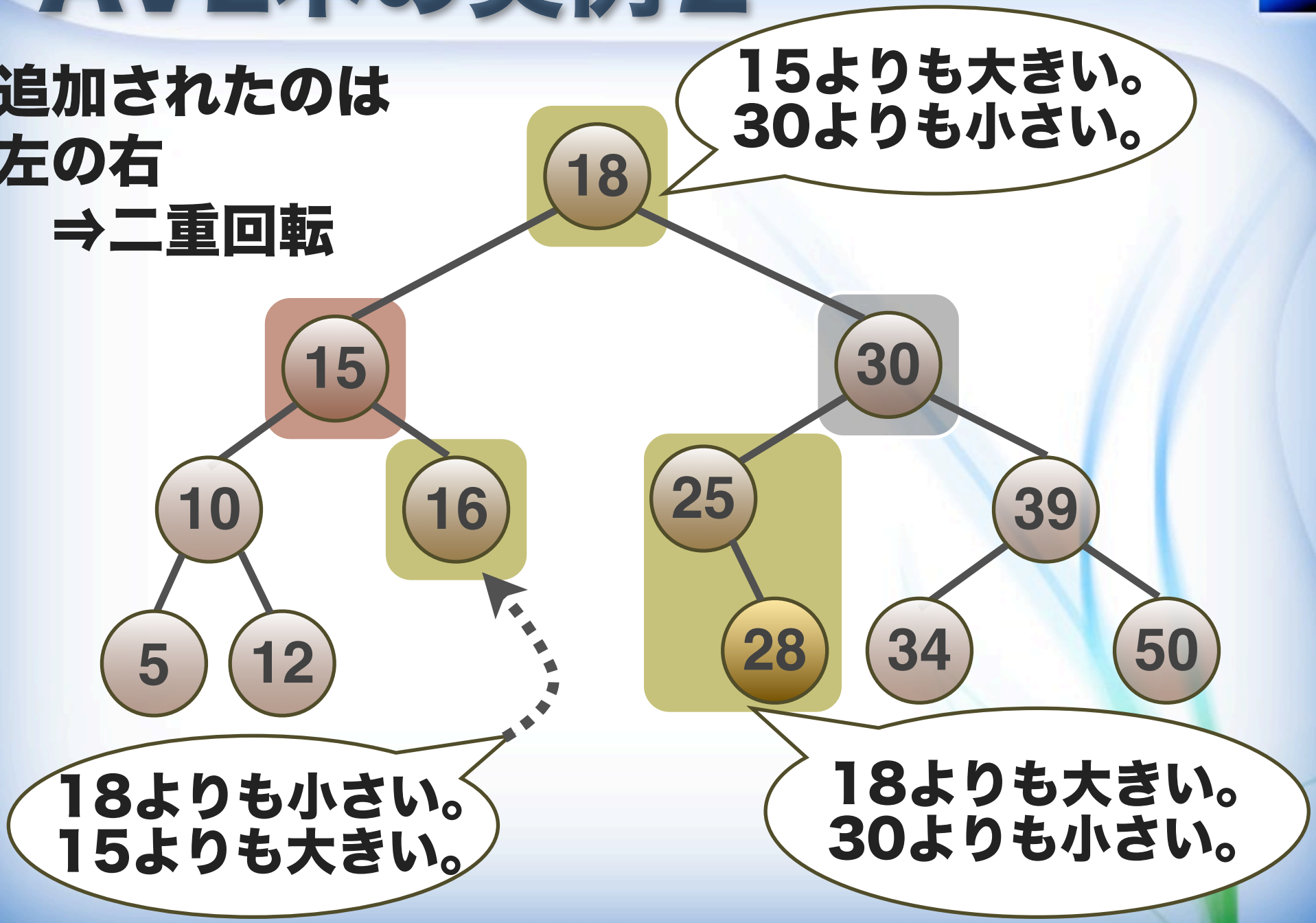
AVL木の実例２

追加されたのは
左の右
⇒二重回転



AVL木の実例２

追加されたのは
左の右
⇒二重回転



AVL木の考察

- 「右の左」と「右の右」への追加は、左の場合の逆（線対称）。
- 削除する場合も追加と同様に高さをチェックする。
- 追加・削除のコストは、追加・削除の部分にまず $O(\lg n)$。整形のために高さを計算するところで、かなり大きめの $O(\lg n)$。
  整形自体は単純な操作なので $O(1)$。

$$O(\lg n) + O(\lg n) + O(1) = O(\lg n)$$

レポート課題

- 次のAVL木に45を追加する様子を書け。
- またその結果から77、92、30の順に削除する様子を書け。