

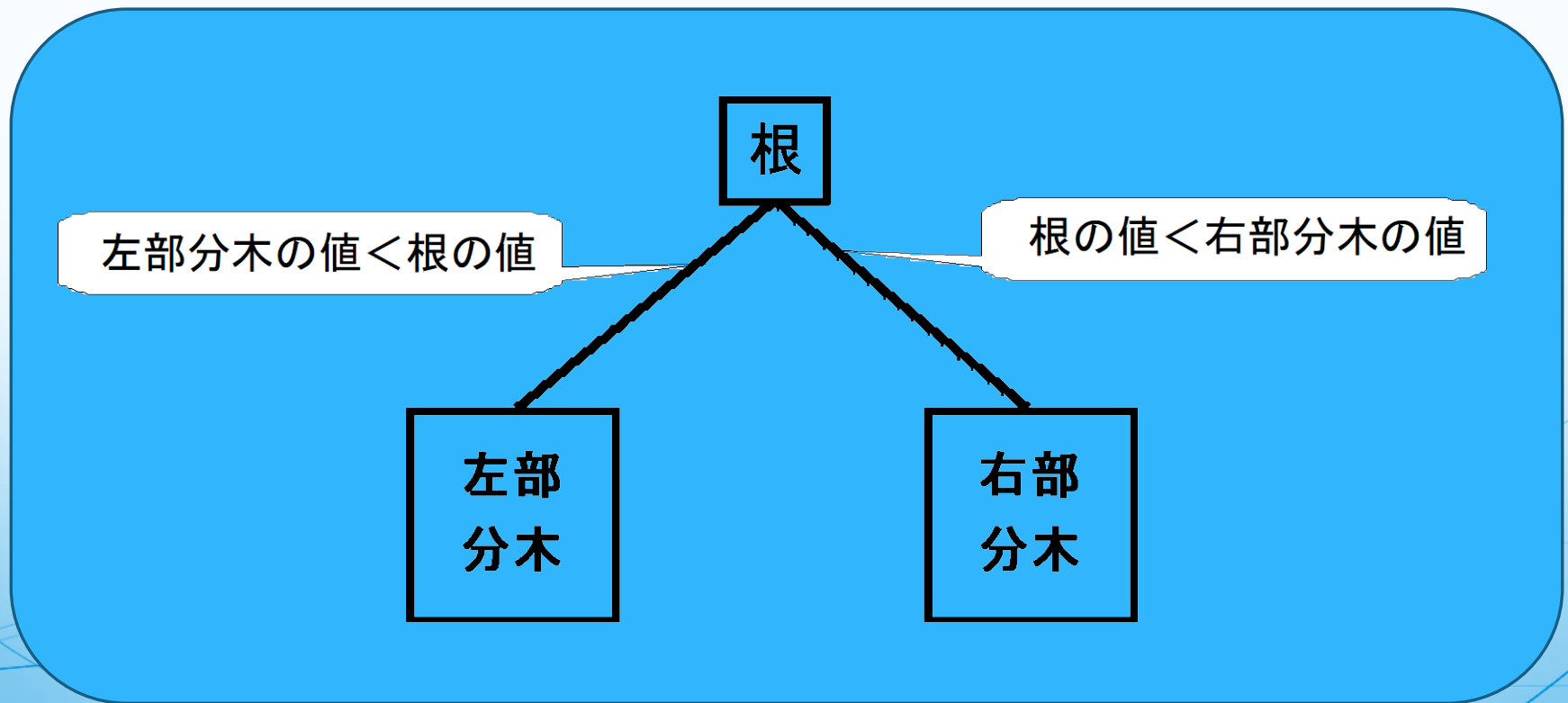
二分探索木

二分探索木の条件

① 任意の根とその部分木について、

- ① 左部分木に含まれる各要素は
根の要素よりも小さく。
- ② 右部分木に含まれる各要素は
根の要素よりも大きい。

② 二分探索木の図



二分探索木の探索

① 二分探索木を利用した探索

- ① 根から枝分かれした二分探索木の
データ構造を利用する。
- ② 二分探索木の条件に従って
データの探索を行う。

② 二分探索木の条件に従った探索アルゴリズム

- ① 各節にデータを置く
- ② 根から葉に向かって辿る経路

二分探索木の探索時間

- ① 探索時間は、枝分かれしているレベルの高さに依存する。
- ② 探索木のレベルが高いほど探索に時間がかかる。
- ③ 二分探索木ではレベルが低いほど効率がよいことになる。

二分探索木の探索手順

- ① 探索データをKとする。
- ② 根の値とKを比較する。
- ③ 根の値=Kならば、探索成功で処理を終了する。
- ④ 根の値>Kならば、
探索データは左部分木にあるので
左の節を次の根と考えて②に戻る。

⑤ 根の値 $<K$ ならば、
探索データは右部分木にあるので
右の節を次の根と考えて②に戻る。

⑥ 比較対象となる根が存在しなくなると、
該当する探索データが存在しないことになり、
探索不成功で処理を終了する。

二分探索木の挿入手順

- ① 探索は二分探索木の根から始める。
- ② 探索データの値と節の値を比較する。
- ③ 探索データと節の値が一致すると、
挿入データが挿入済みであるので処理を終了する。
- ④ 探索データが節の値よりも小さいとき、
左の節に移動し、節があれば②に戻る。

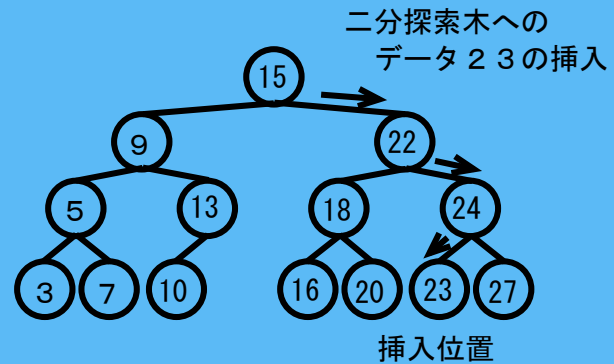
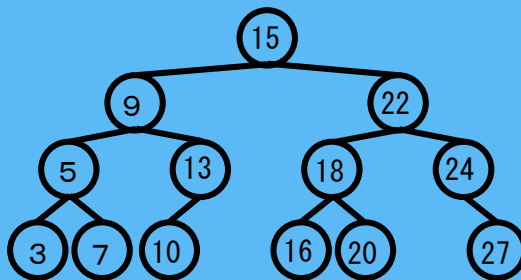
- 
- ⑤ 探索データが節の値よりも大きいとき、
右の節に移動し、節があれば②に戻る。
 - ⑥ 探索すべき節または葉が存在しなくなると、
その位置に新しいデータを挿入し、処理を終了する。
- 

二分探索木挿入の具体例

二分探索木に23を挿入する手順は次のように実行する。

- ① 15と23を比較して、 $15 < 23$ であるから右の経路を進む。
- ② 22と23を比較して、 $22 < 23$ であるから右の経路を進む。
- ③ 24と23を比較して、 $23 < 24$ であるから左の経路を進む。
- ④ 24の左の葉が存在しないためその位置に23を挿入する。

二分探索木挿入の図

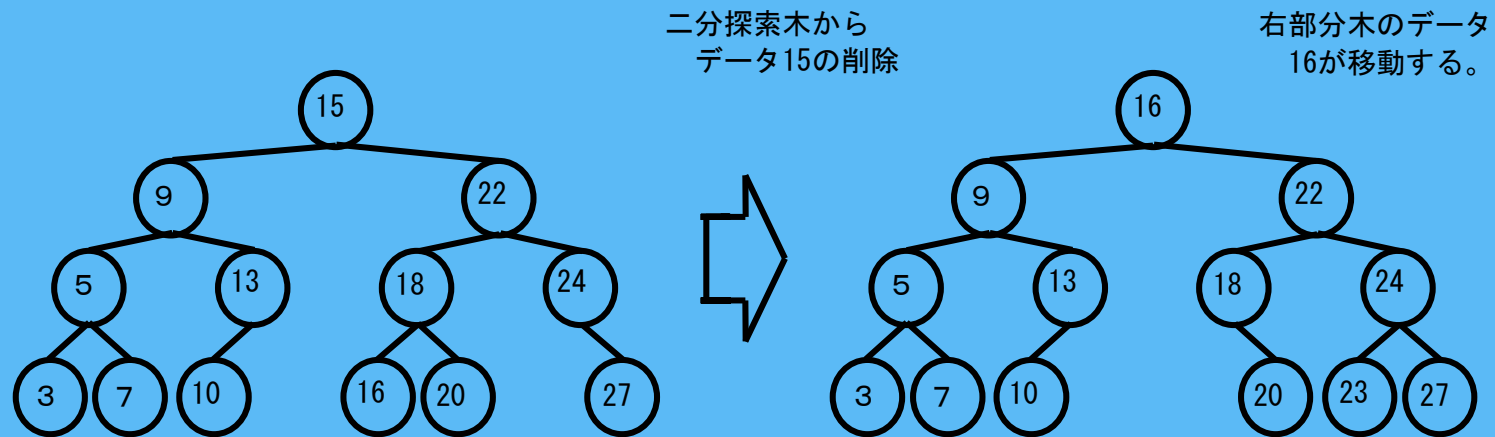


二分探索木のデータの削除手順

- ① 二分探索木を用いて探索し、
削除すべきデータを見つけ削除する。
- ② 削除した節が子を持たないときは何もしない。
- ③ 削除した節が子を一つ持っているとき、
その子を親の代わりにする。

- ④ 削除した節が子を二つ持つ場合は、
右部分木に含まれる最小のデータ
または左部分木に含まれる最大のデータを
親の代わりとして移動する。
- ⑤ 移動した子が更に子を持つときは、
移動した子が削除されたものとして、
同様の処理を繰り返す。

二分探索木削除の図



左部分木のデータ13が移動することも可能である。