

データ構造・アルゴリズム論

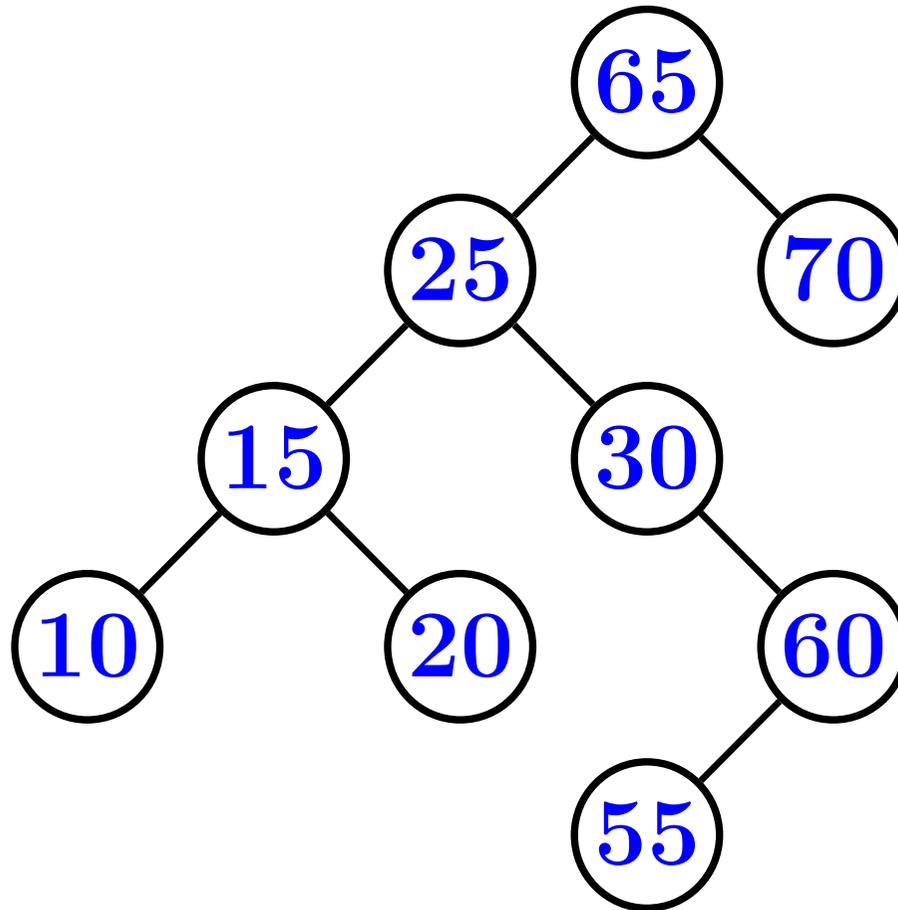
2分探索木の操作手順

山田 俊行

<https://www.cs.info.mie-u.ac.jp/~toshi/>

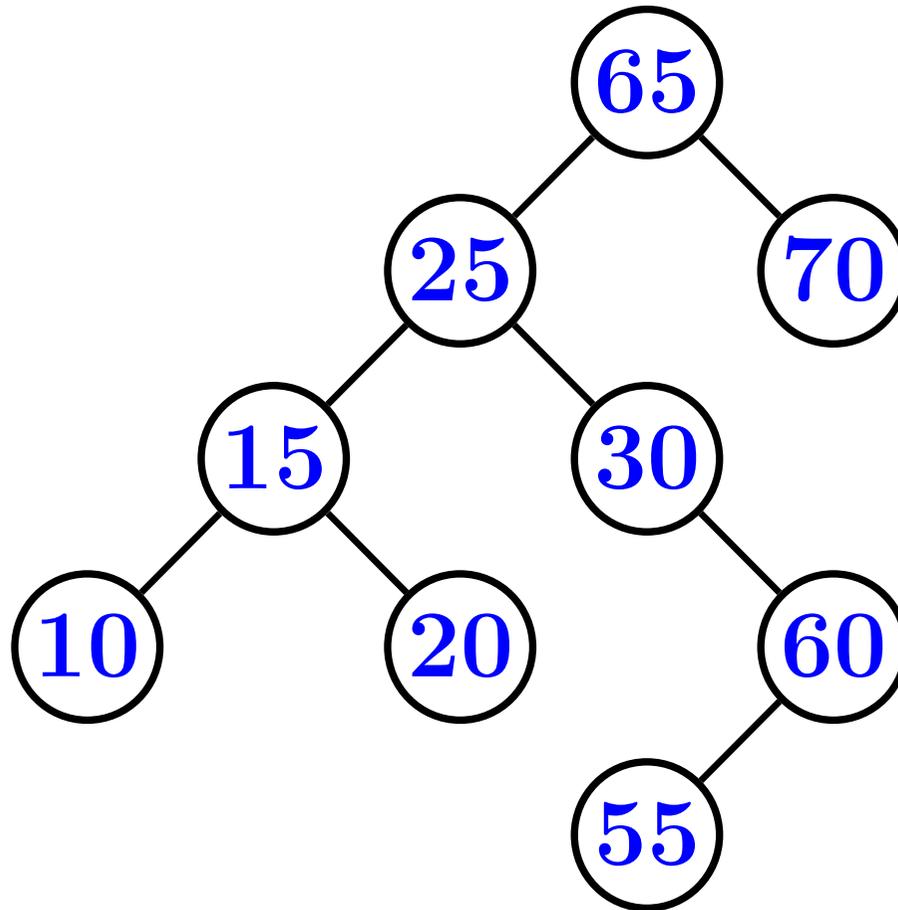
2分探索木

2分探索木条件
左子孫 < 頂点 < 右子孫



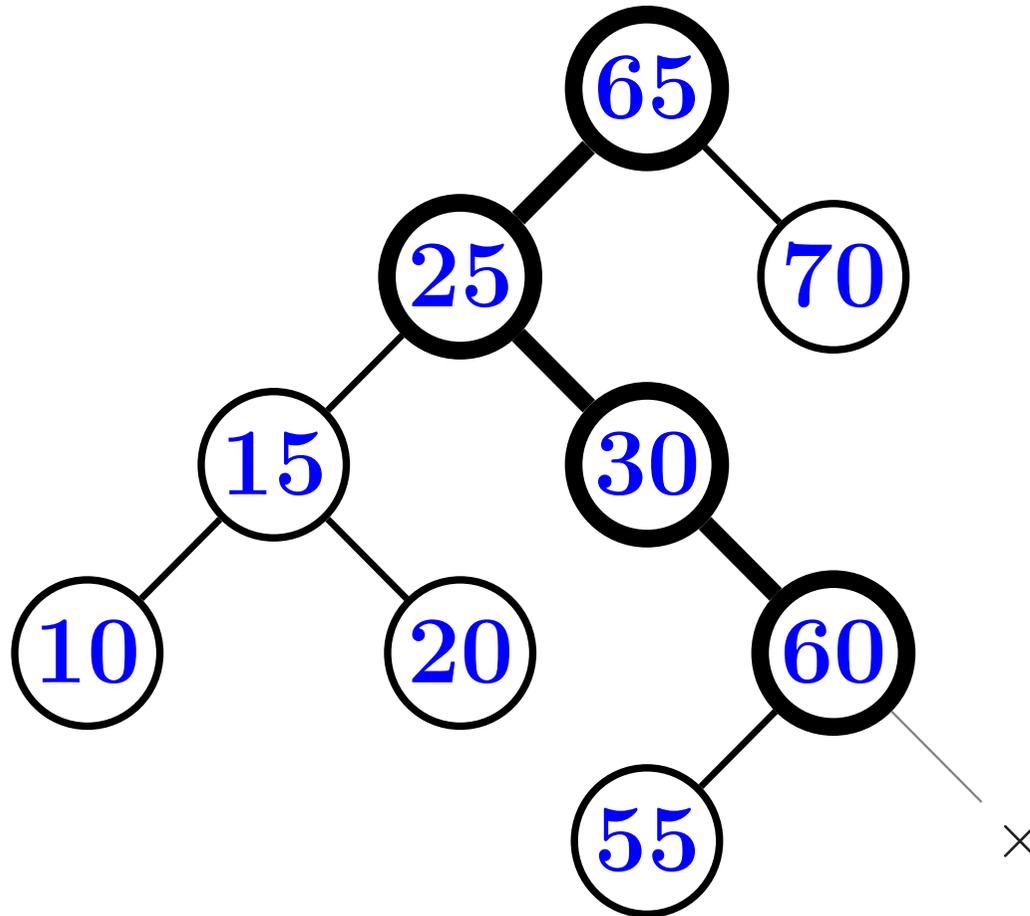
2分探索木への挿入

要素 63 の挿入前



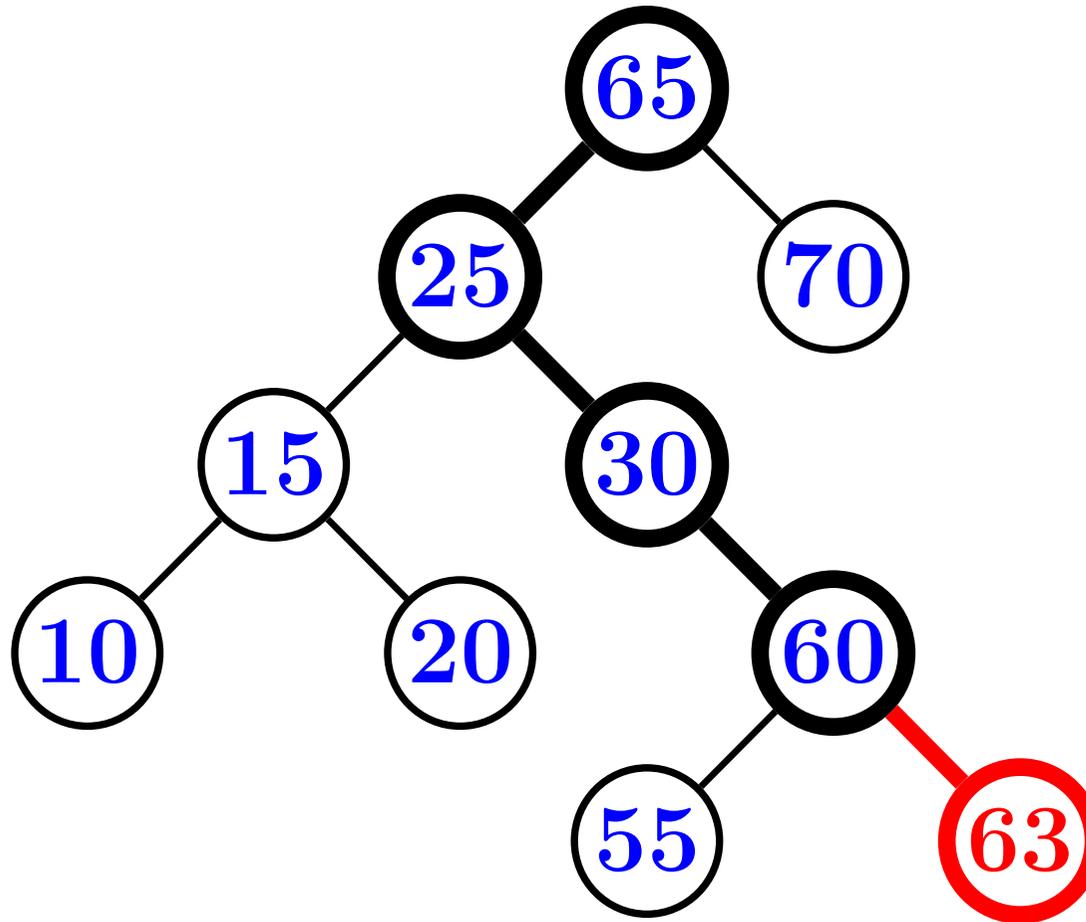
2分探索木への挿入

2分探索の失敗位置に葉を追加



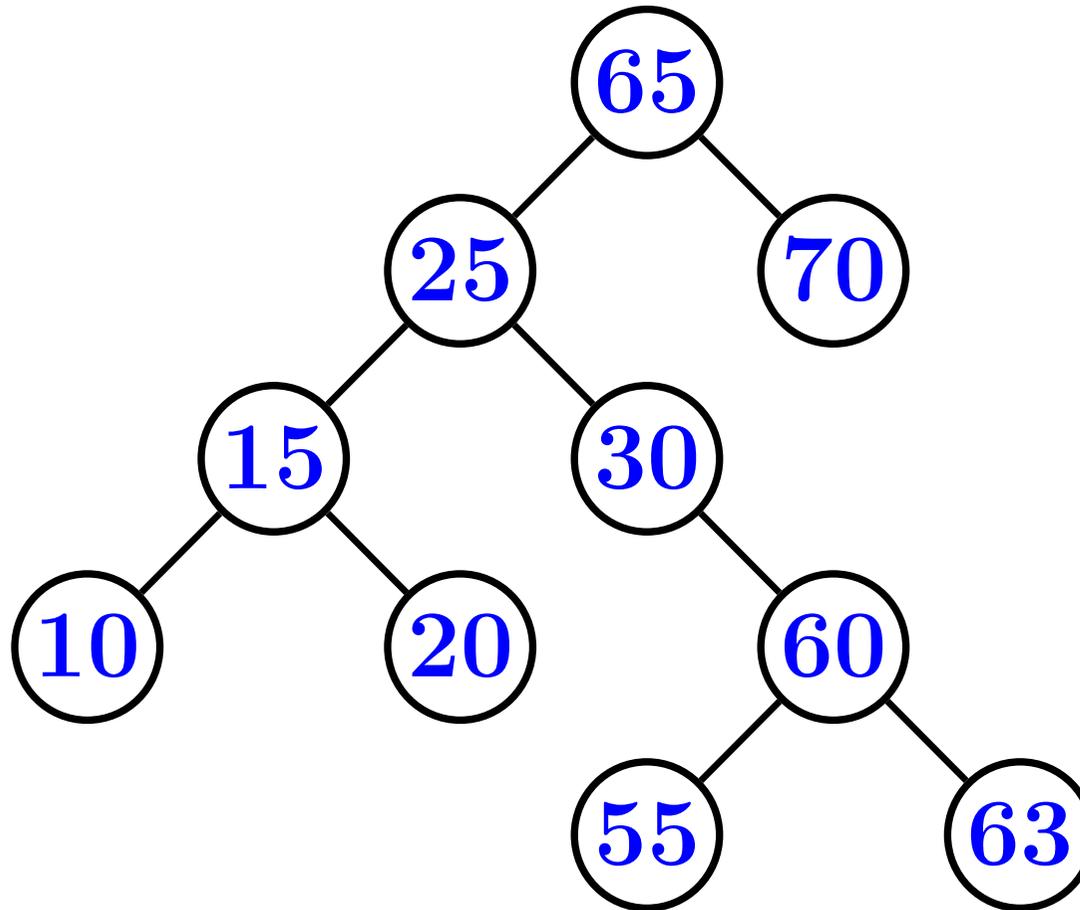
2分探索木への挿入

2分探索の失敗位置に葉を追加



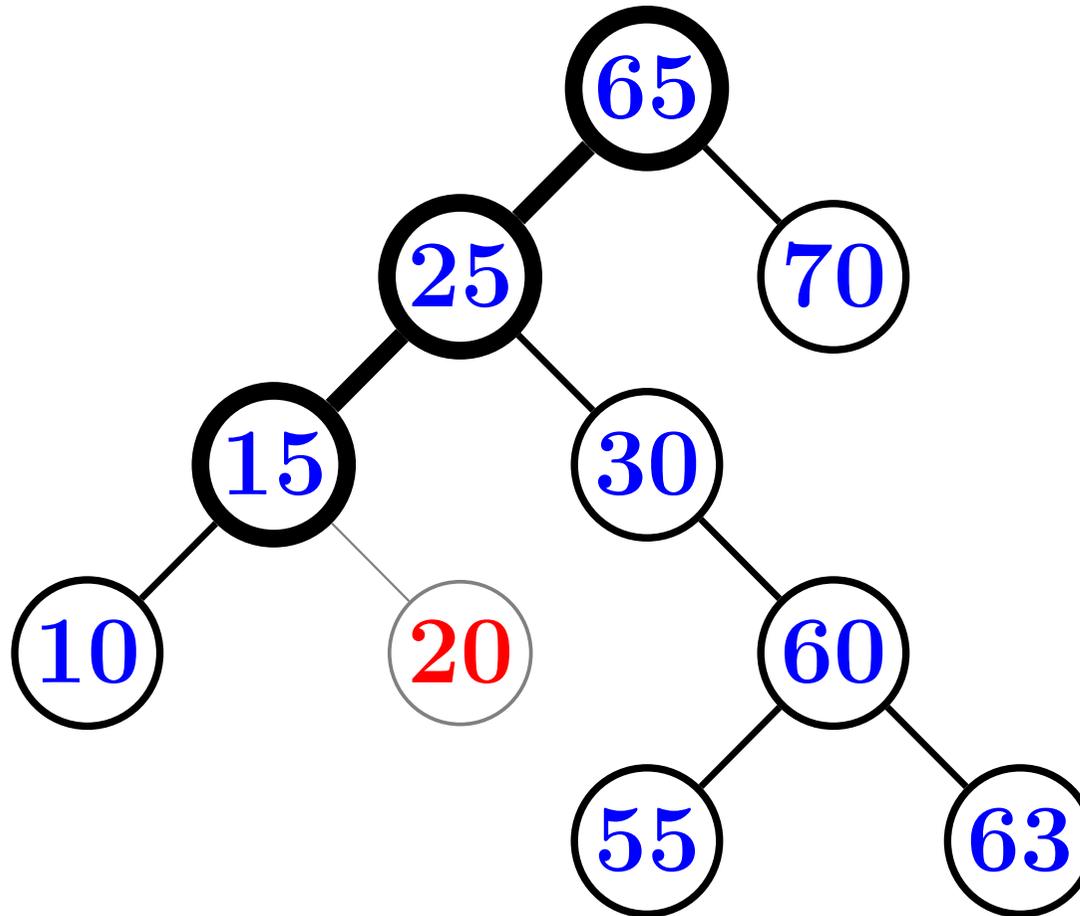
2分探索木からの削除

要素 20 の削除前



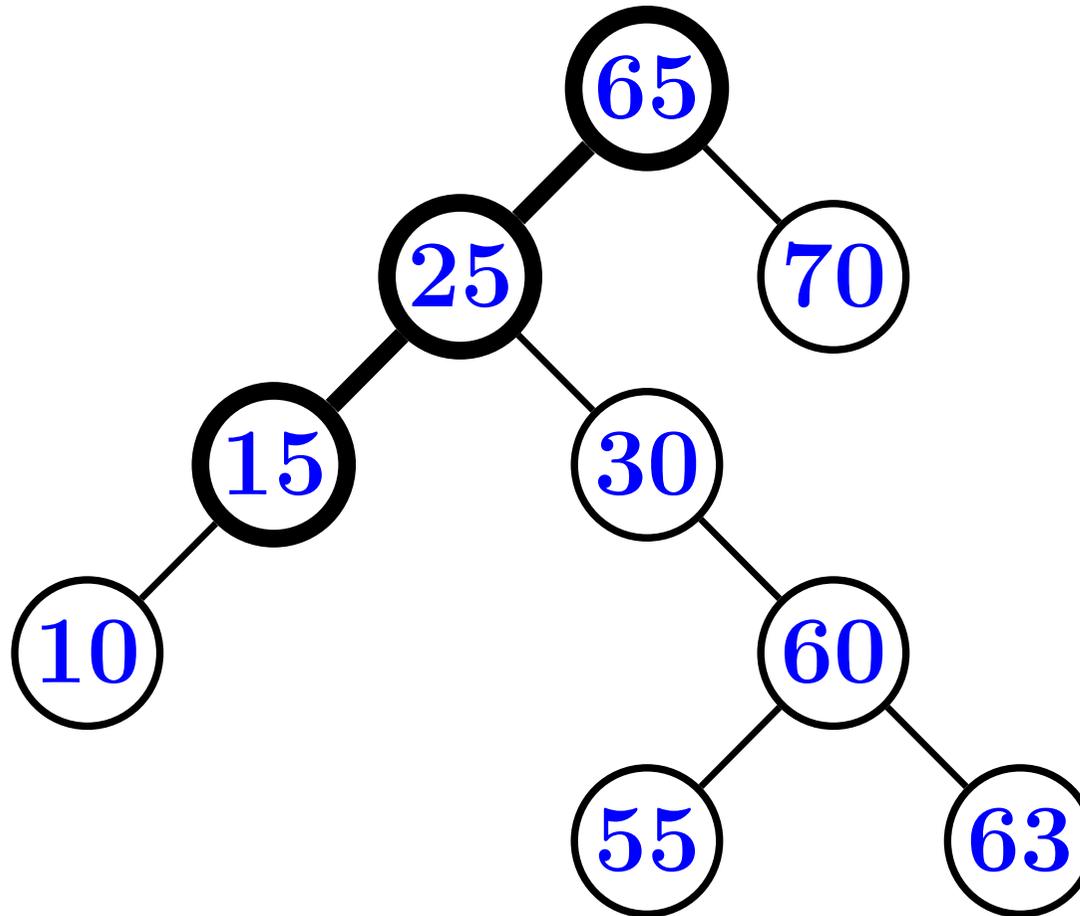
2分探索木からの削除

2分探索して、葉なら、それを削除



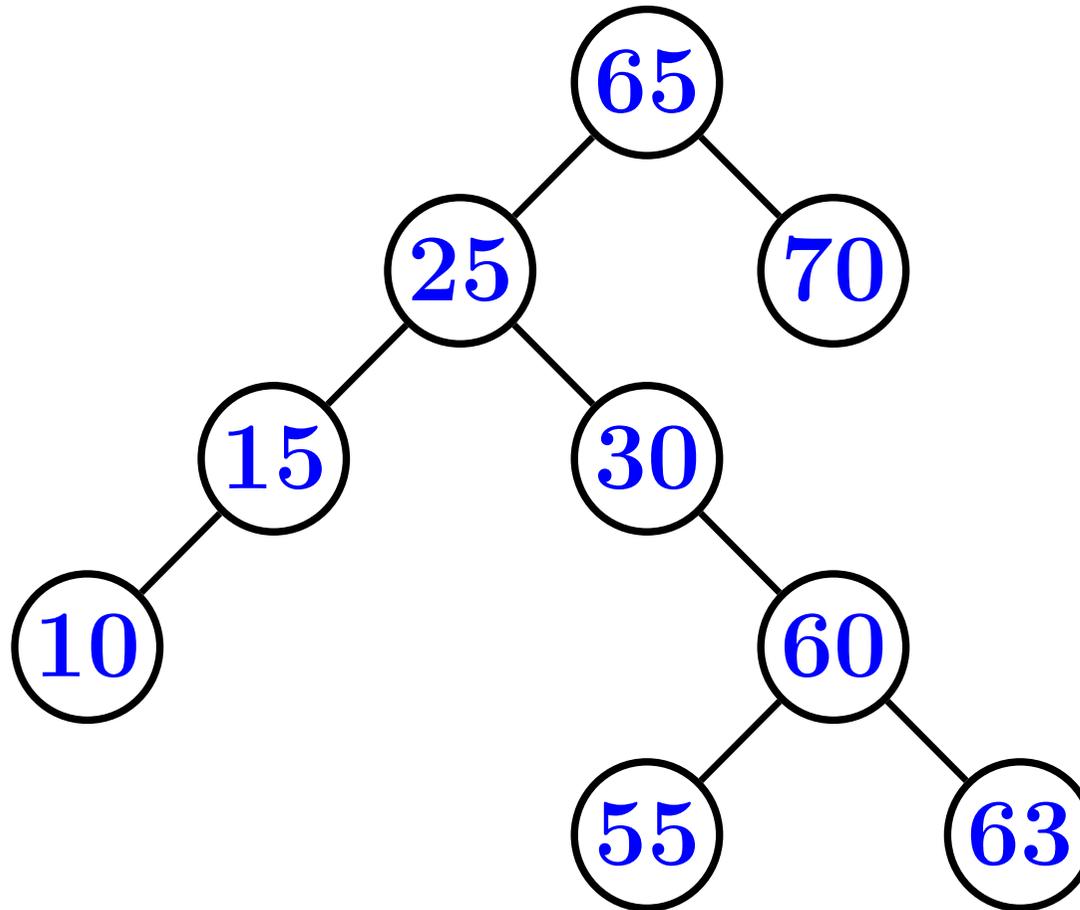
2分探索木からの削除

2分探索して，葉なら，それを削除



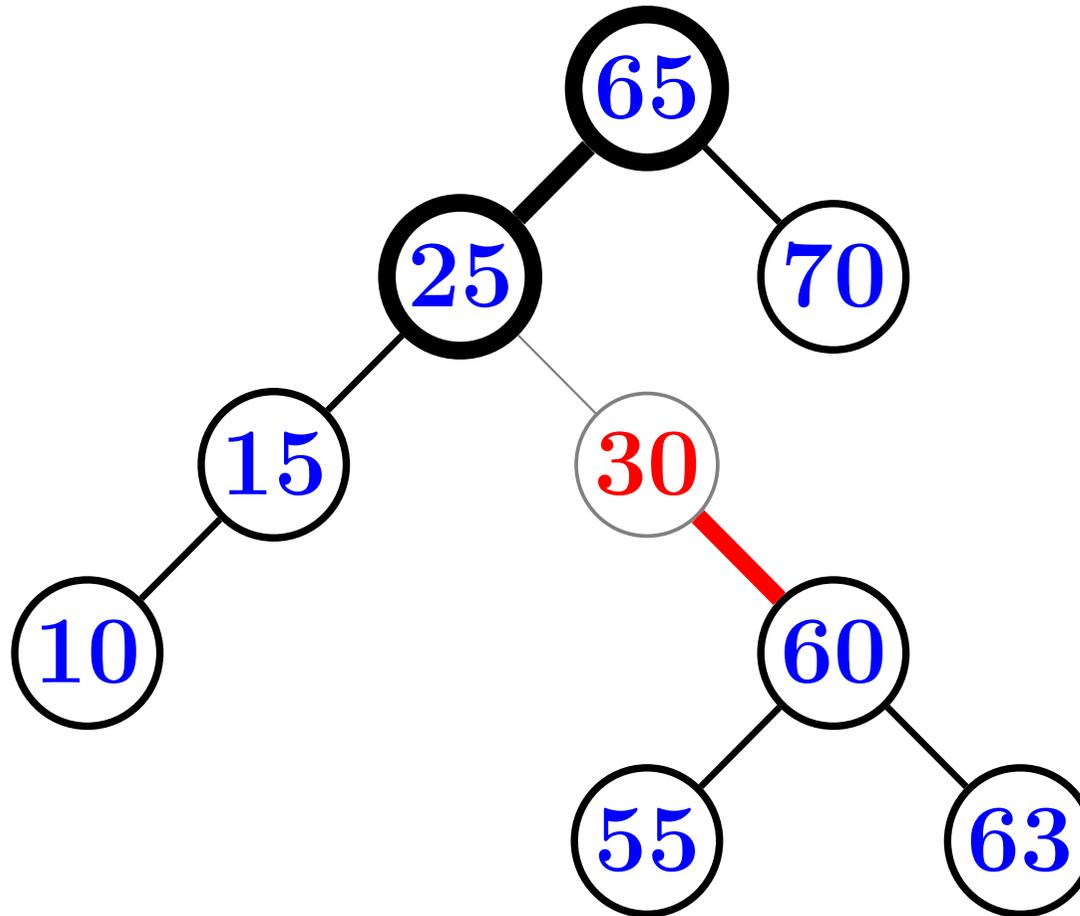
2分探索木からの削除

要素 30 の削除前



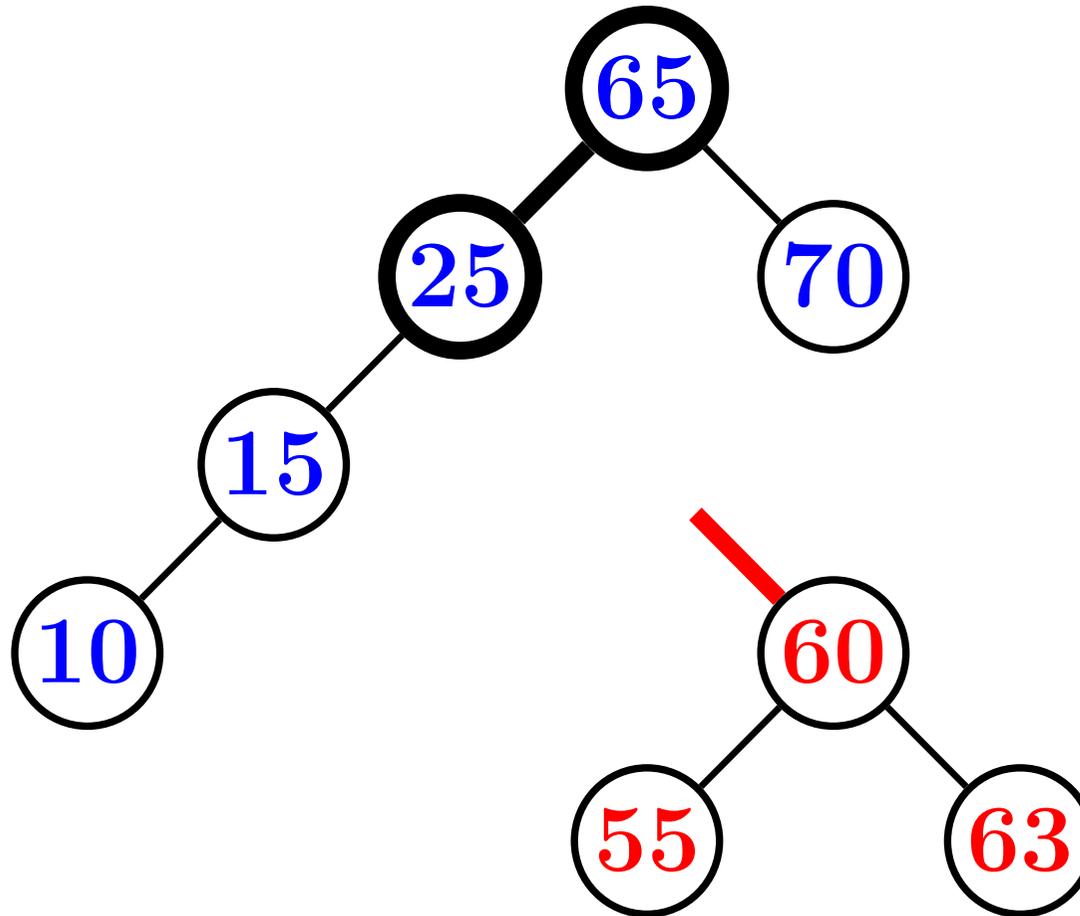
2分探索木からの削除

2分探索して、子が一つなら、削除位置に部分木を連結



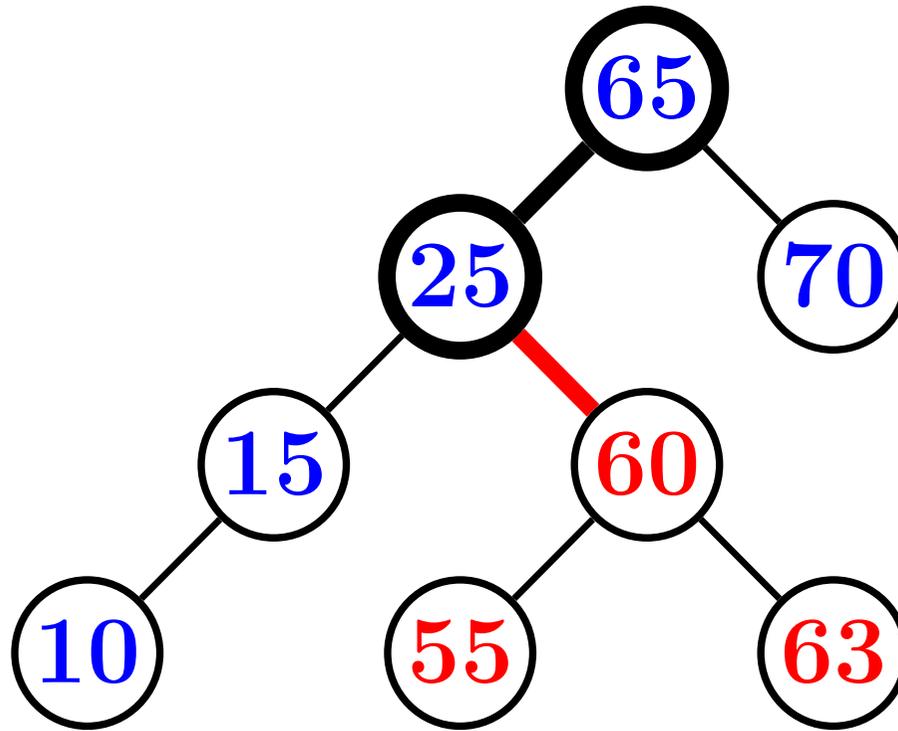
2分探索木からの削除

2分探索して、子が一つなら、削除位置に部分木を連結



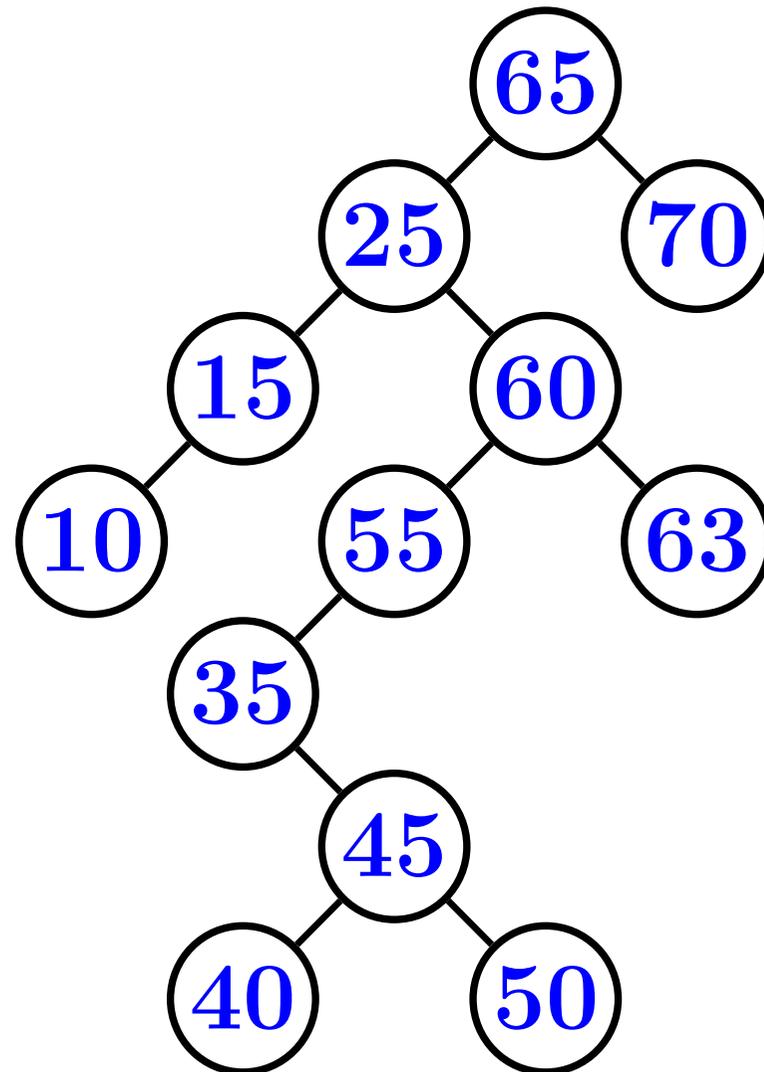
2分探索木からの削除

2分探索して、子が一つなら、削除位置に部分木を連結



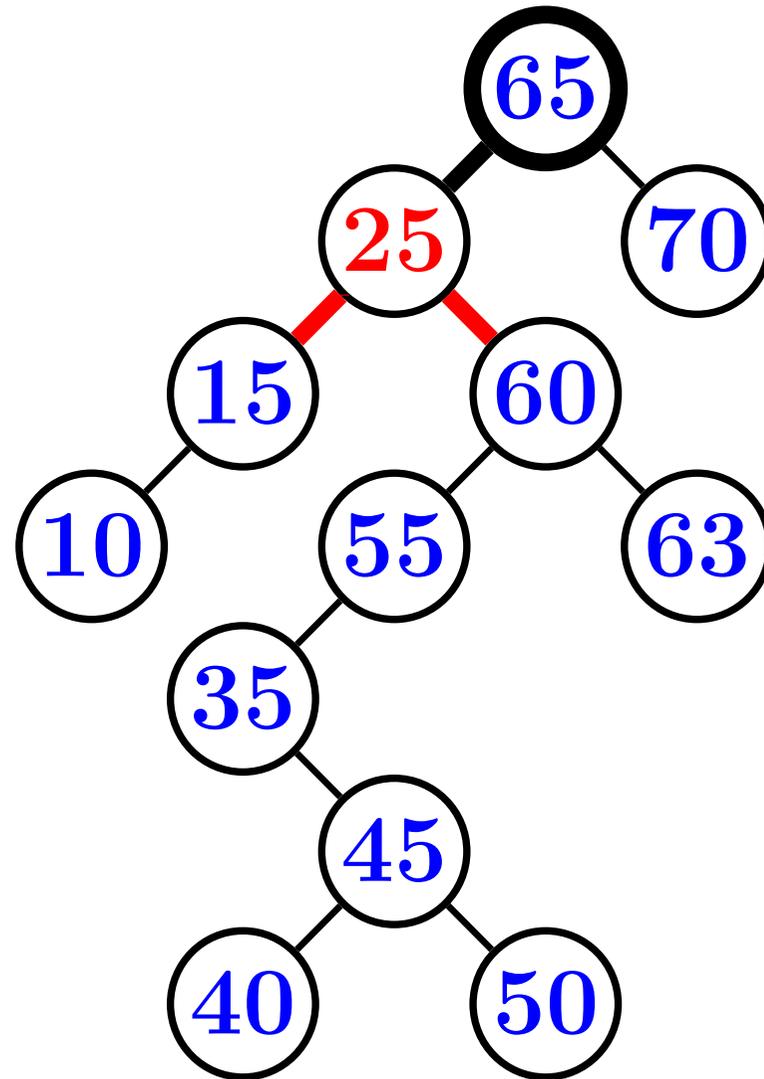
2分探索木からの削除

要素 25 の削除前



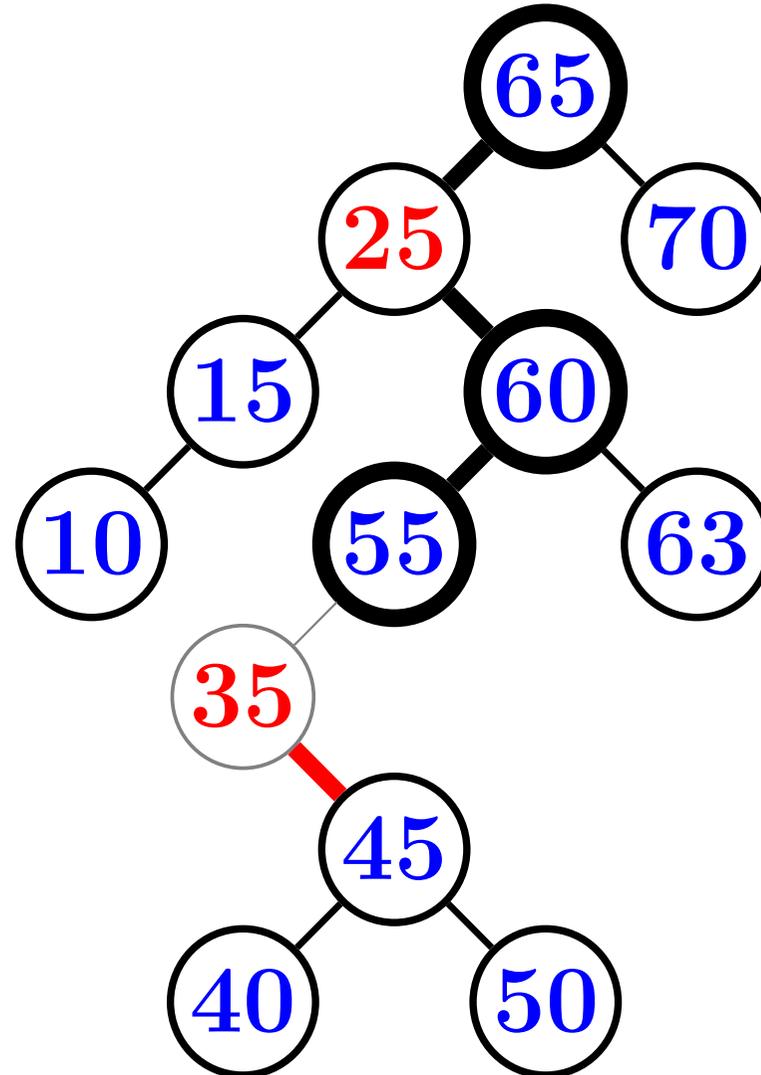
2分探索木からの削除

2分探索して、子が二つなら



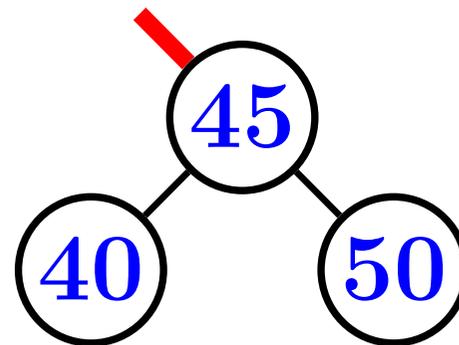
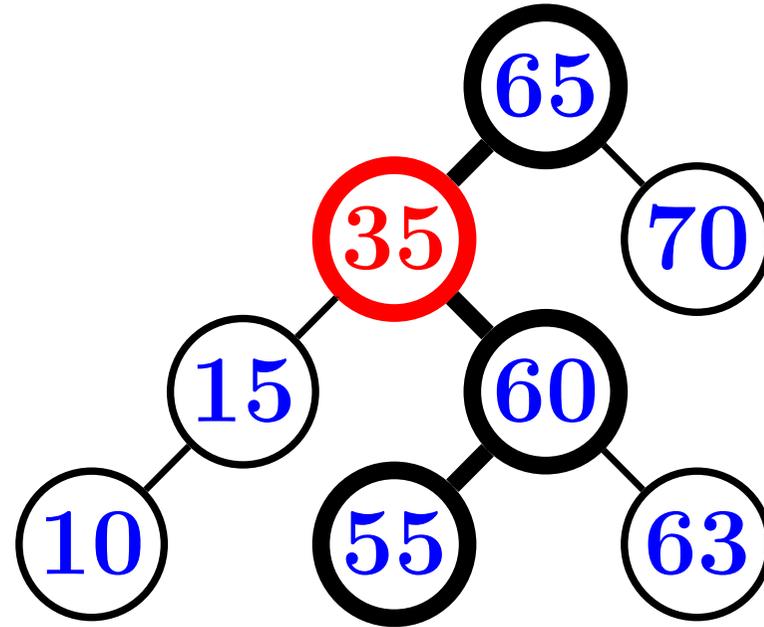
2分探索木からの削除

1. 削除する値の位置に，次に大きい値を移動



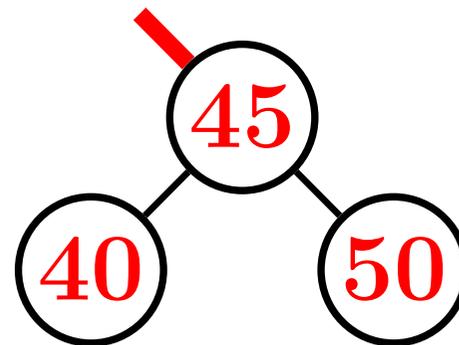
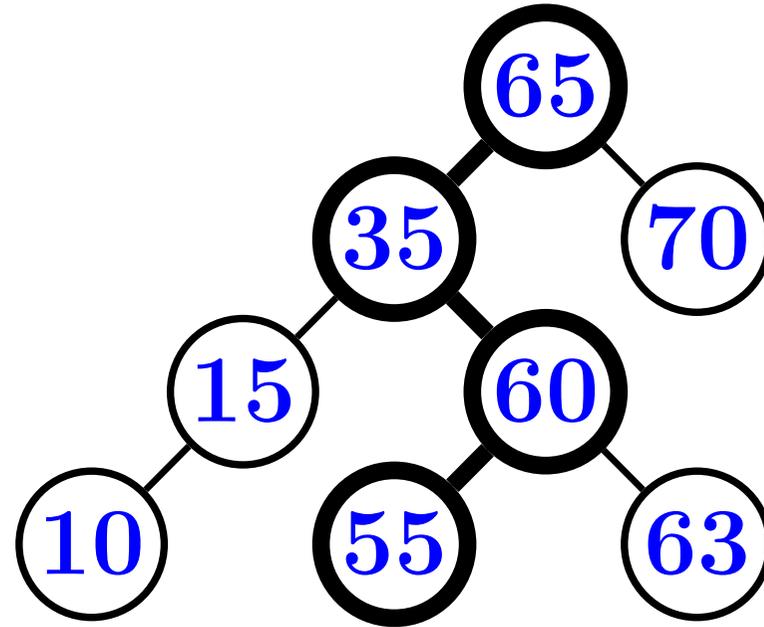
2分探索木からの削除

1. 削除する値の位置に，次に大きい値を移動



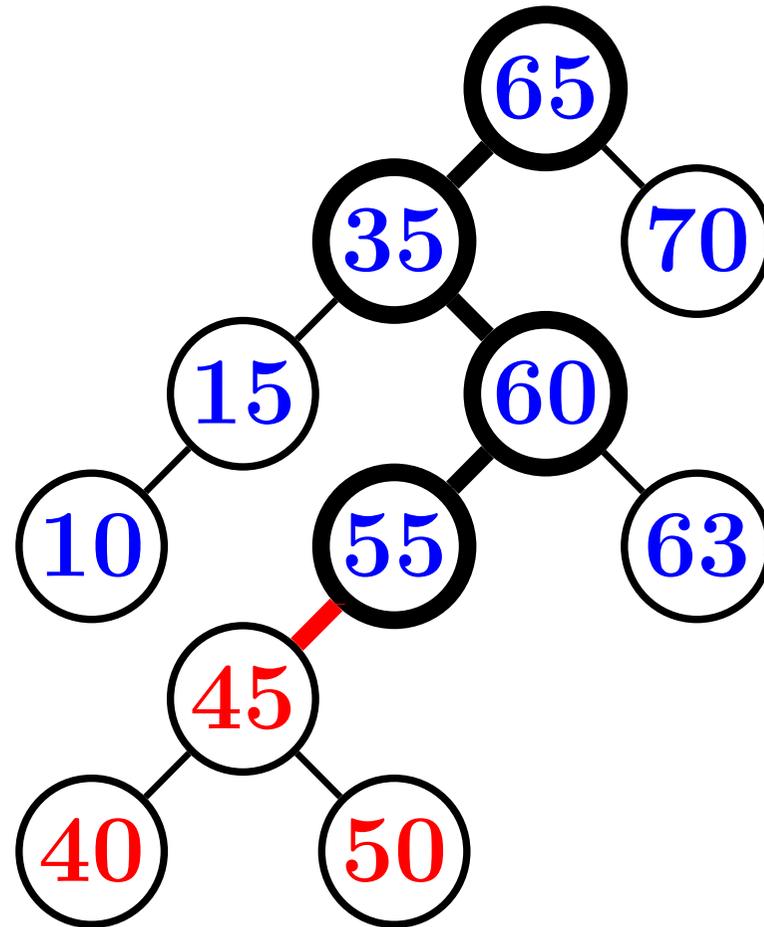
2分探索木からの削除

2. 空いた位置に部分木を連結



2分探索木からの削除

2. 空いた位置に部分木を連結



2分探索木の操作手順

挿入

2分探索して，探索の失敗位置に葉を追加

削除

2分探索して，

- 葉なら，それを削除
- 子が一つなら，削除位置に部分木を連結
- 子が二つなら，
 1. 削除する値の位置へ，次に大きい値を移動
 2. 空いた位置に部分木を連結