

アルゴリズム論 (第11回)



岩手県立大学
Iwate Prefectural University

佐々木研(情報システム構築学講座)

講師 山田敬三

k-yamada@iwate-pu.ac.jp

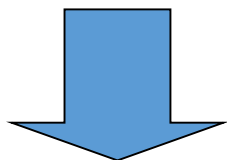
2分木



岩手県立大学
Iwate Prefectural University

2分探索のためのデータ構造

- 連結リストの探索には、線形探索が使われる
- 線形探索は2分探索に比べて遅い！
- 2分探索を動的なデータ構造に適応することを考える。
 - ⇒ 2分探索木
 - 2分木自体はすでに済



テキストファイル中の単語の出現頻度を調べるプログラム

完全バランス2分木



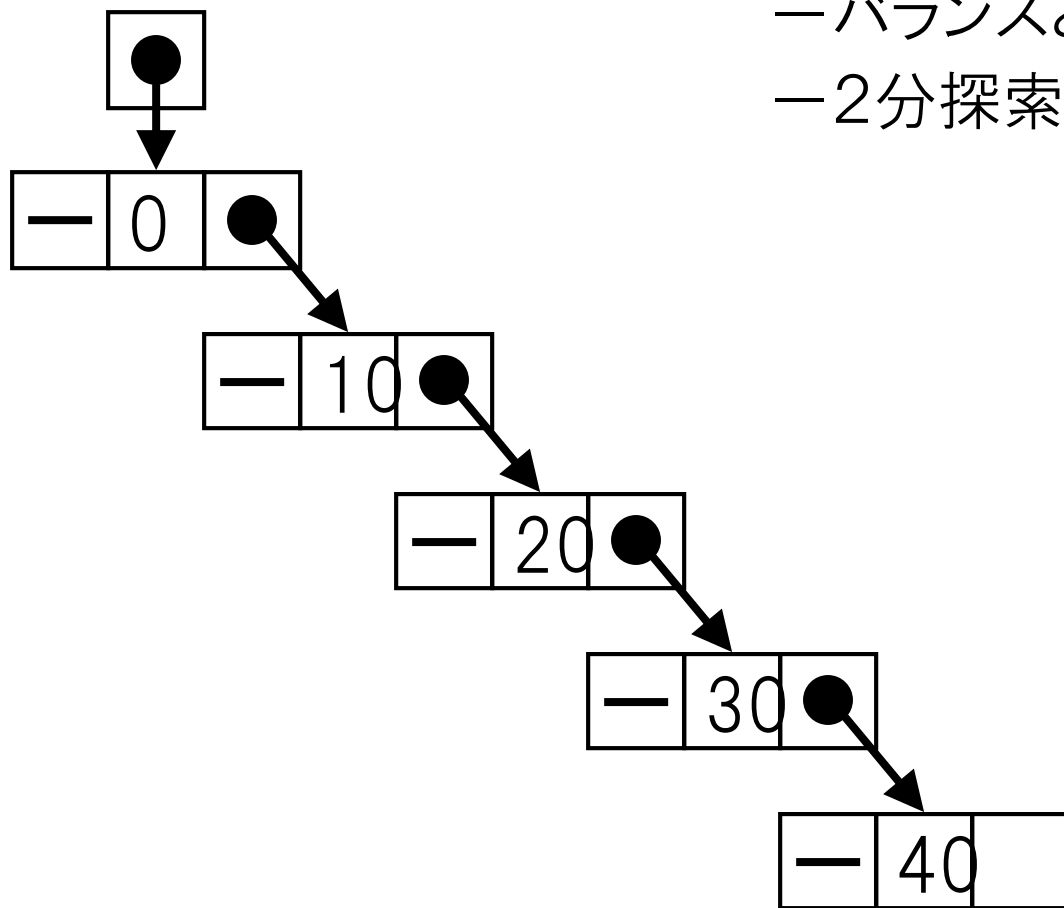
岩手県立大学
Iwate Prefectural University

バランスのよい2分木

✓2分木はバランスしていることが望ましい.

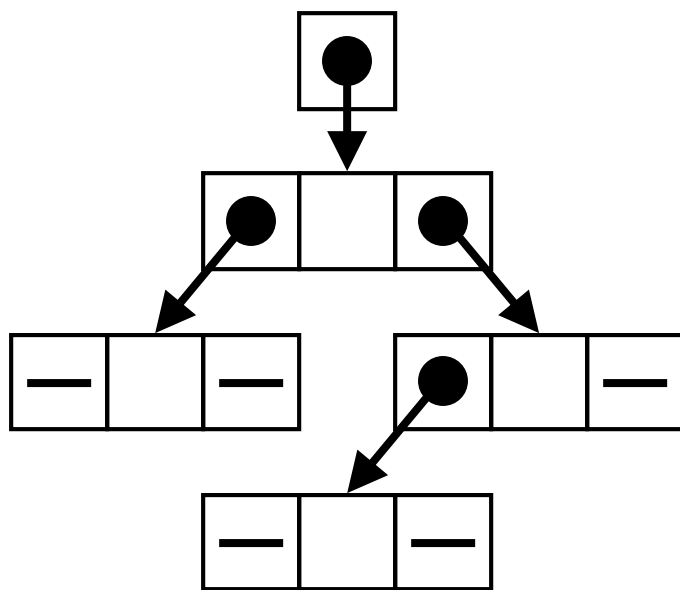
ーバランスとは

ー2分探索木と連結リストとの比較

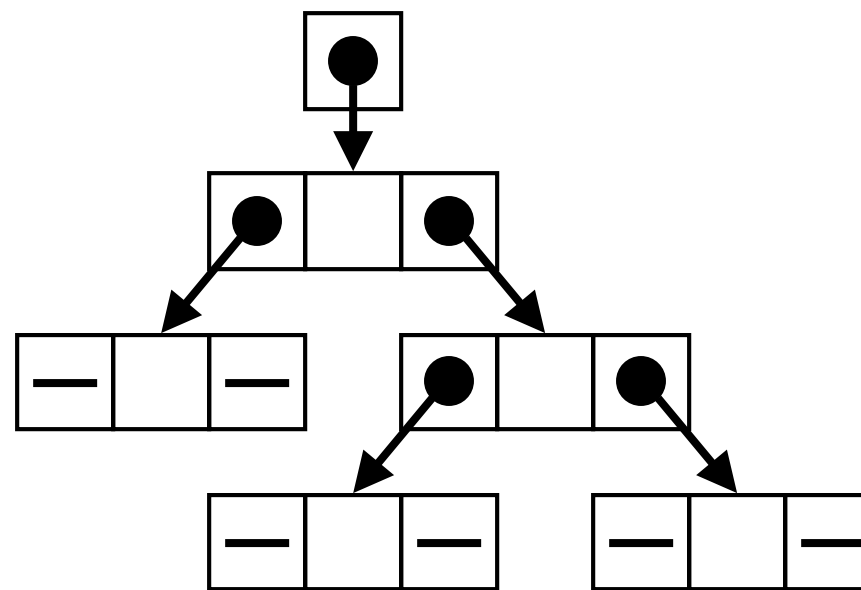


完全にバランスした木

- ✓ 全てのノードにおいて、その左部分木と右部分木でそれぞれのノードの数がたかだか1つしか変わらない木.



完全にバランス



高さがバランス(AVL木)

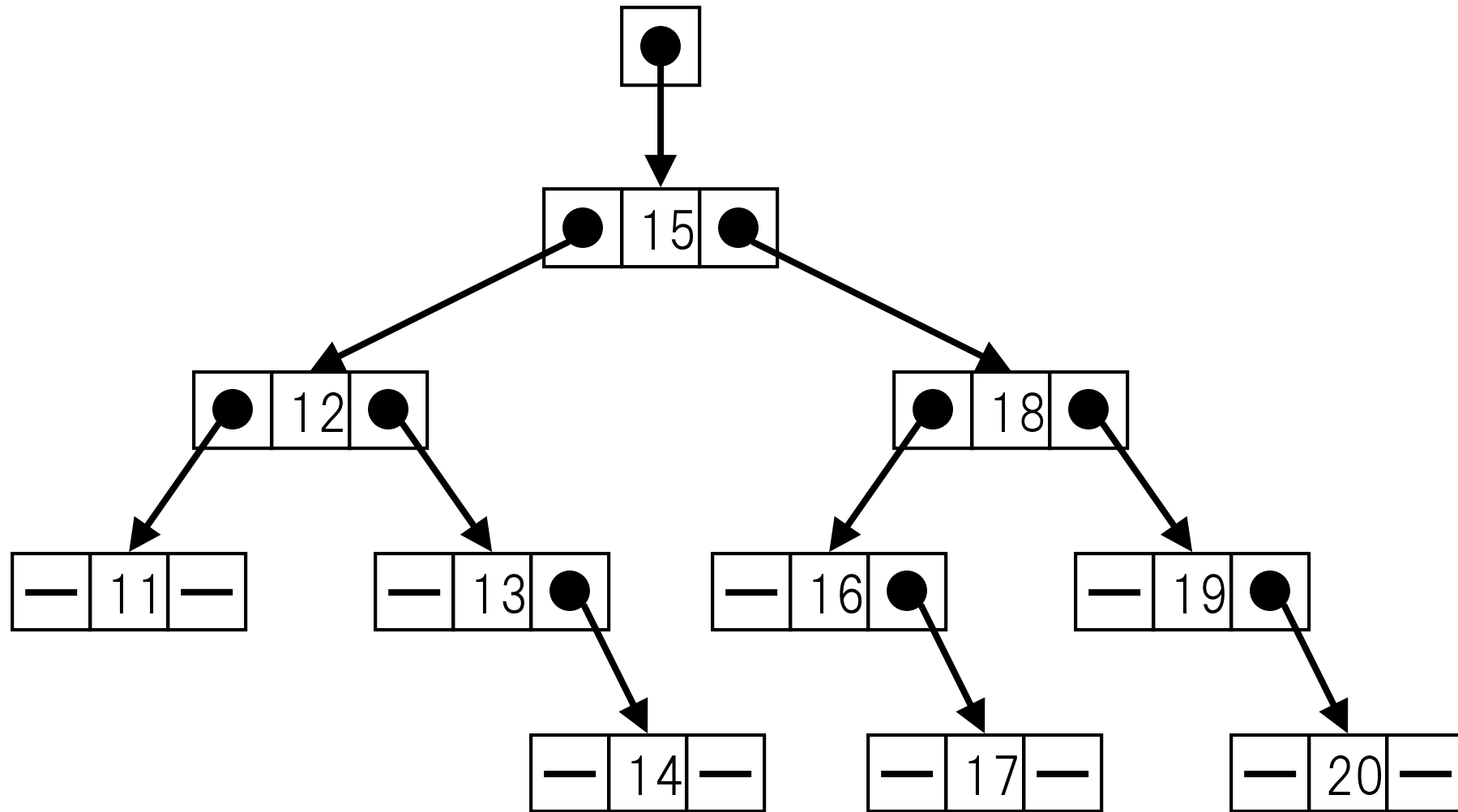
変換

- ✓2分木から「完全にバランスした2分木」への変換
- ✓読み込むデータが昇順で与えられている.
- ✓データの数があらかじめわかっている.

```
1: node* pbtree(int n) {
2:     int nleft, nright, nleftplusright = n - 1;
3:     node* p;
4:     if (n == 0) { return NULL; }
5:     nleft = nleftplusright / 2;
6:     nright = nleftplusright - nleft;
7:     p = (node*)malloc(sizeof(node));
8:     p->left = pbtree(nleft);
9:     scanf("%d", &p->num);
10:    p->right = pbtree(nright);
11:    return p;
12: }
```

```
typedef struct Node {
    int num;
    struct Node *left, *right;
} node;
```

2分探索木(データ数:10)



情報システムへの応用

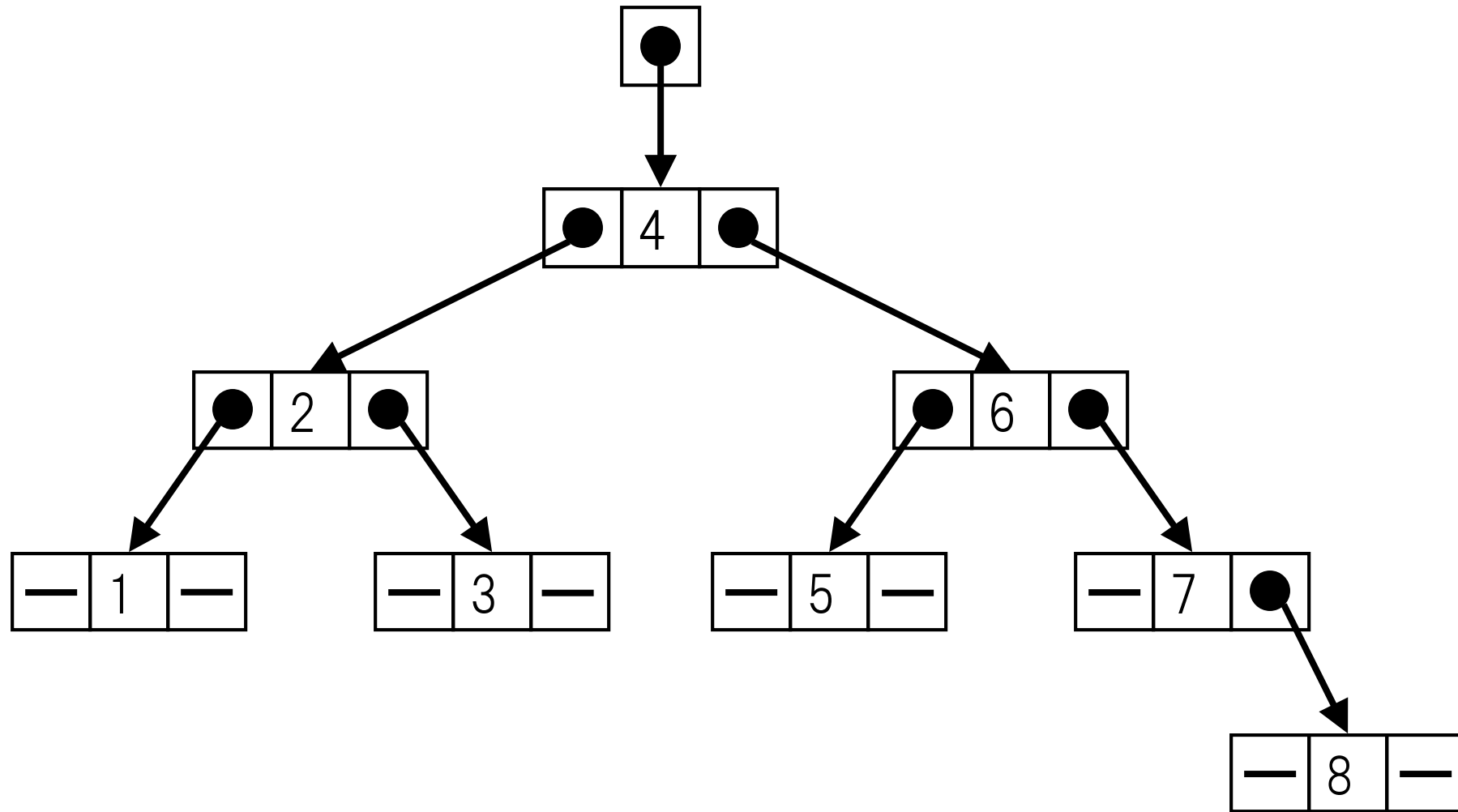
✓データ: 個人名と数値(電話番号, 登録番号など)

1. ファイルから全てのデータを読み込み,
完全にバランスした2分探索木を生成する. (.L)
2. 指定した名前のデータを探索する. (<name> ?)
3. 新しいデータを追加する. (<name> <value>)
4. 指定したデータを削除する. (<name> /)
5. 指定したデータの数値を変更する. (<name> <value>)
6. 全てのデータをファイルにセーブする. (.S)
7. 全てのデータをアルファベット順に表示する. (.P)

問題

- 下記の8個のデータに対して, 関数 pbtree を実行したときの, 完全にバランスした2分木を答えなさい.
 - 1, 2, 3, 4, 5, 6, 7, 8

2分探索木(データ数:8)



B-木



岩手県立大学
Iwate Prefectural University

B-木

- ✓大容量データを処理するためには補助記憶装置が必要.
- ✓全データ中, 必要部分を読込, 利用することを考える.
- ✓補助記憶装置(ハードディスク)のアクセスは遅い.

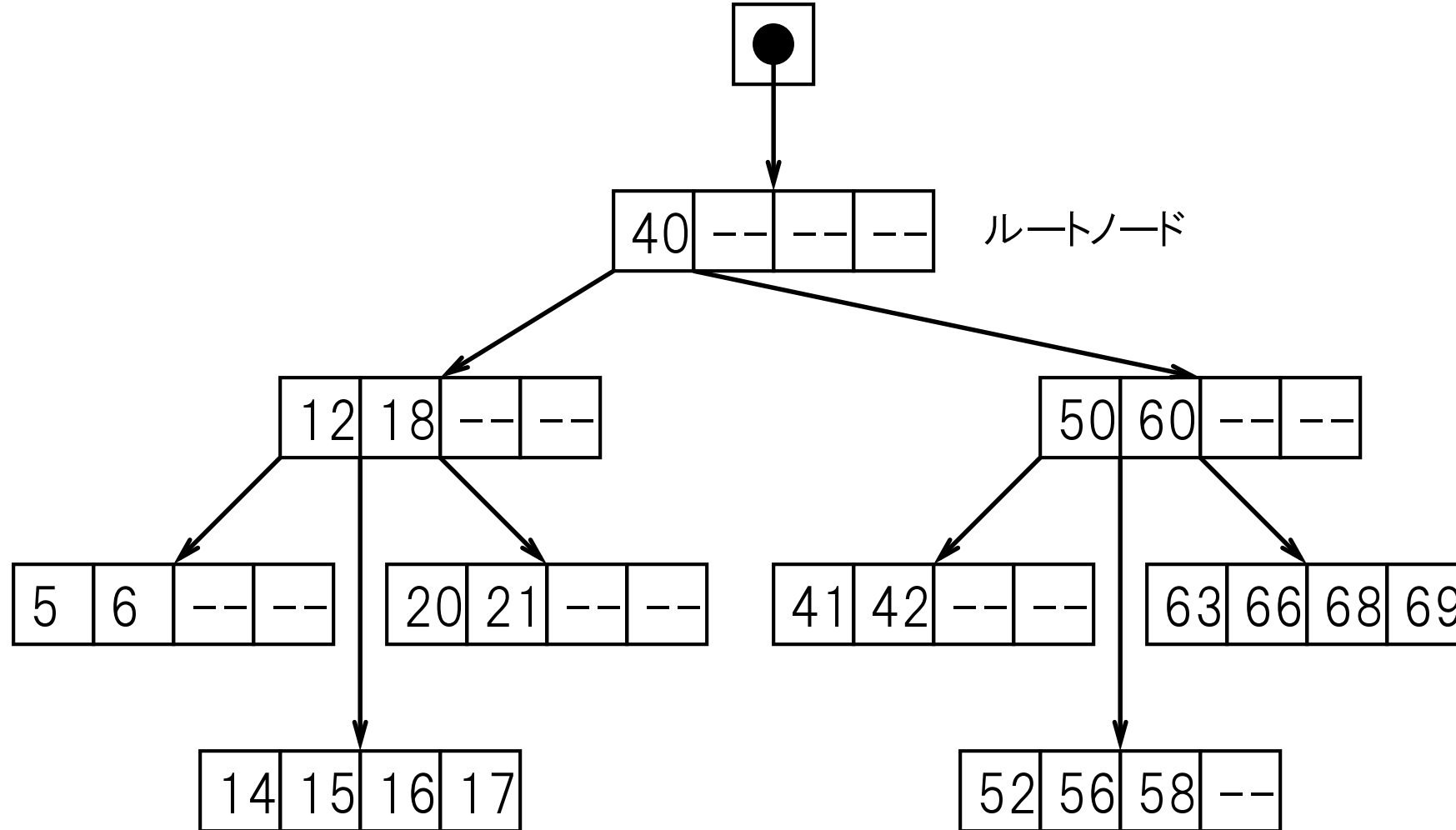
キーを含むデータをグループ化し,
大きなデータブロックとして扱うことを考える.



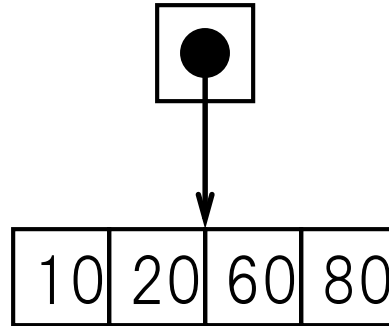
B-木(多分木の種類)

- ✓固定値Mに対して, 各ノードは最大2M個のデータを含む.
- ✓根ノードを除く各ノードは少なくともM個のデータを含む.
- ✓根ノードは少なくとも1個のデータを含む.
- ✓各ノードのデータは昇順に並んでいる.
- ✓B-木の全ての葉は, 同じレベルにある.

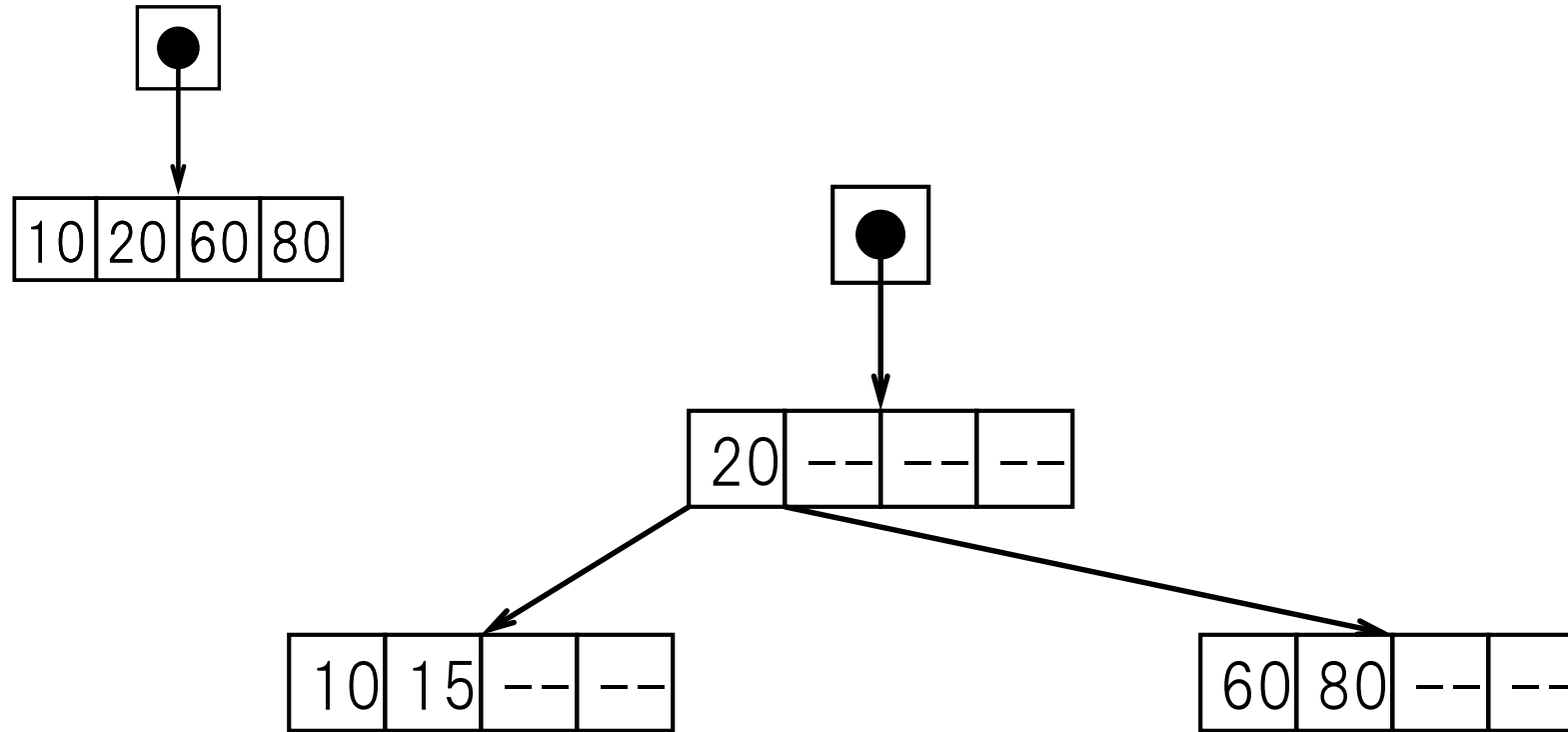
B-木の例（次数：2）



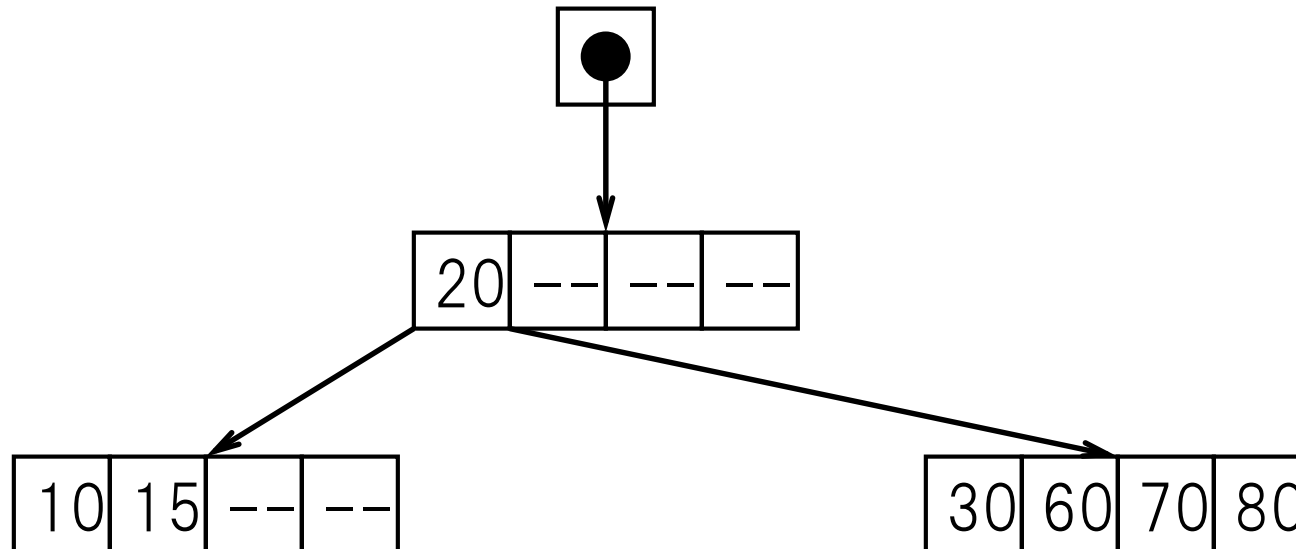
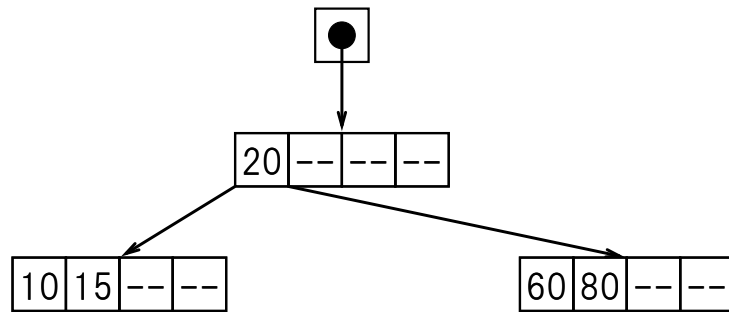
B-木の変化 : 60,20,80,10を挿入



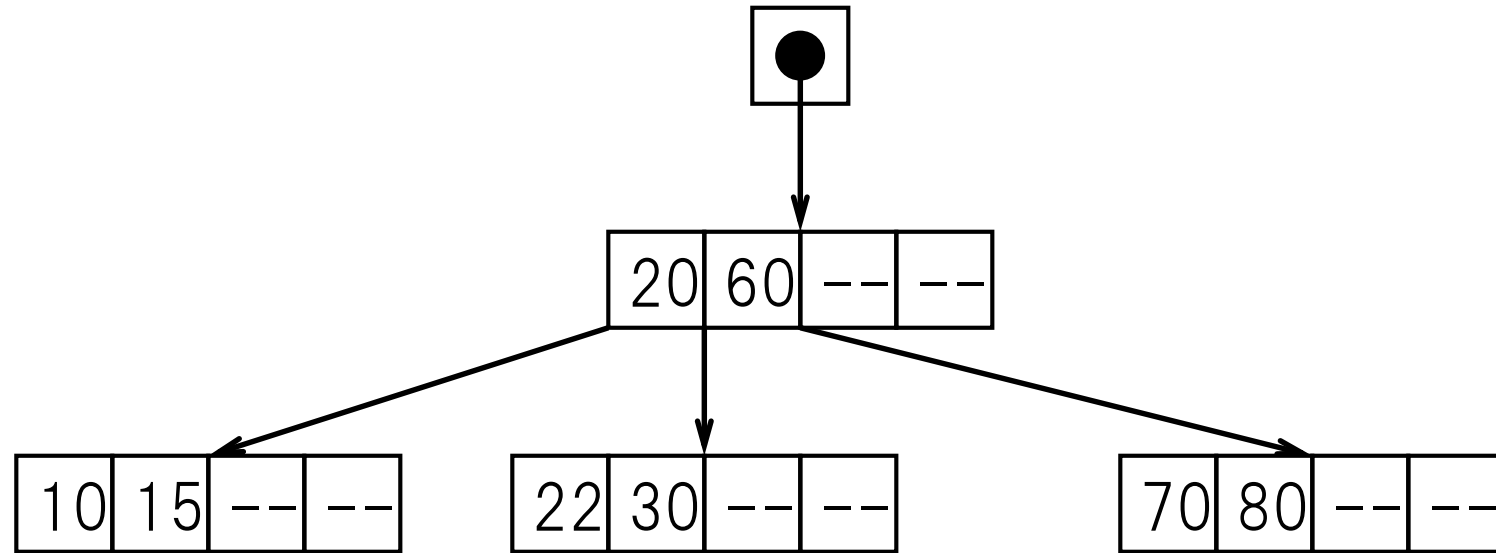
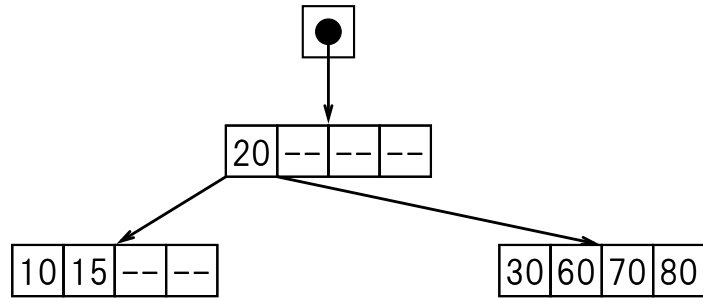
B-木の变化：15を挿入



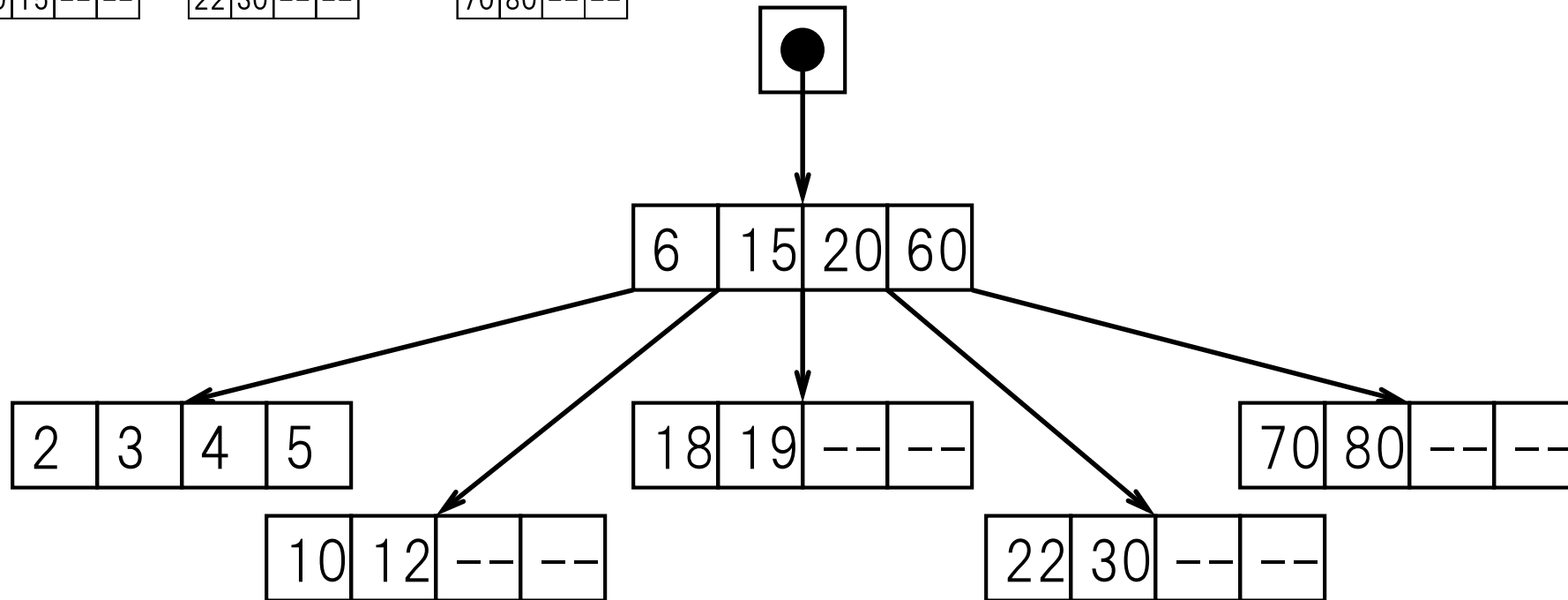
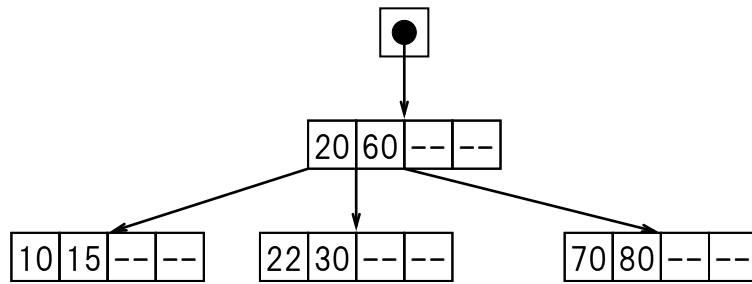
B-木の变化 : 30,70を挿入



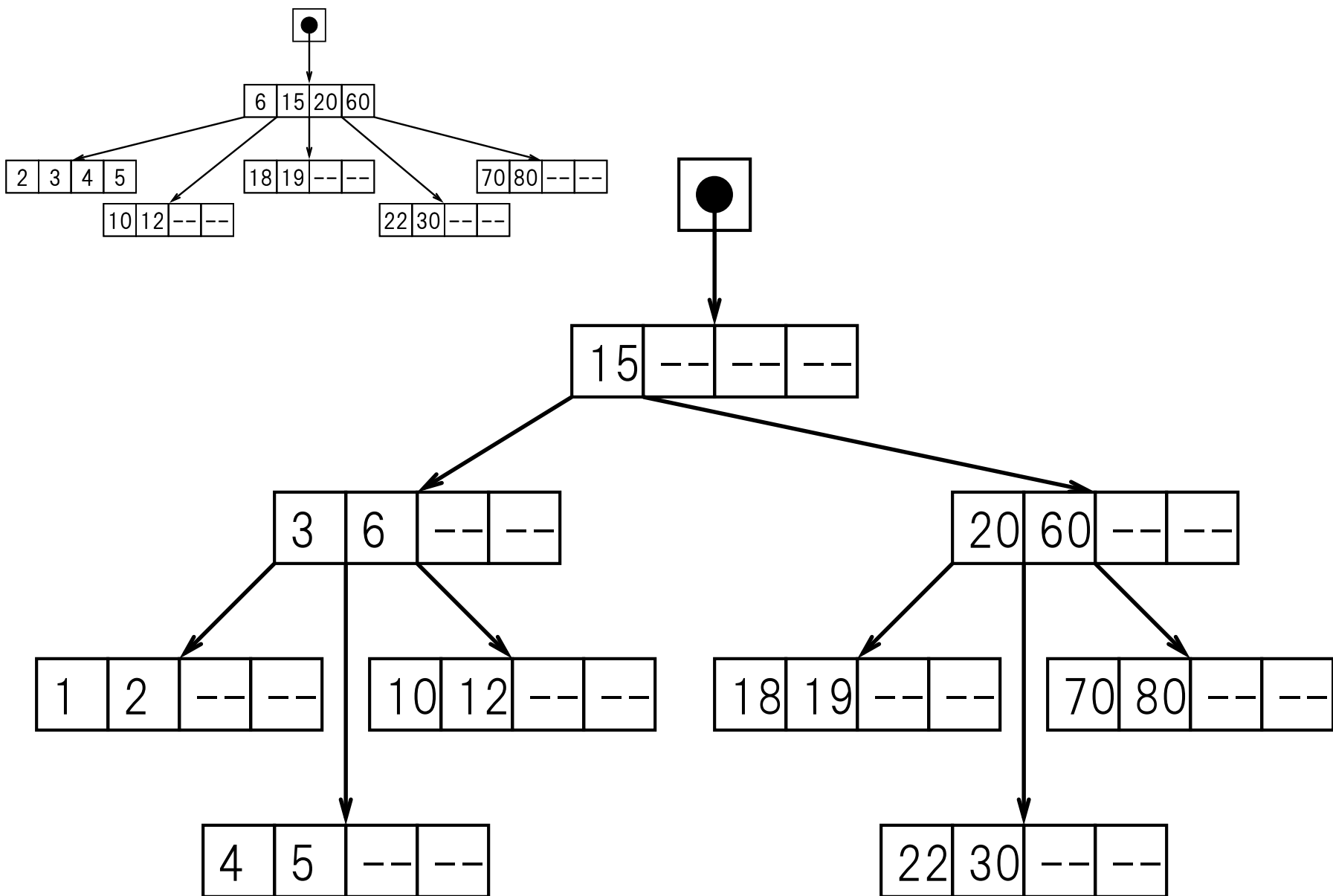
B-木の变化：22を挿入



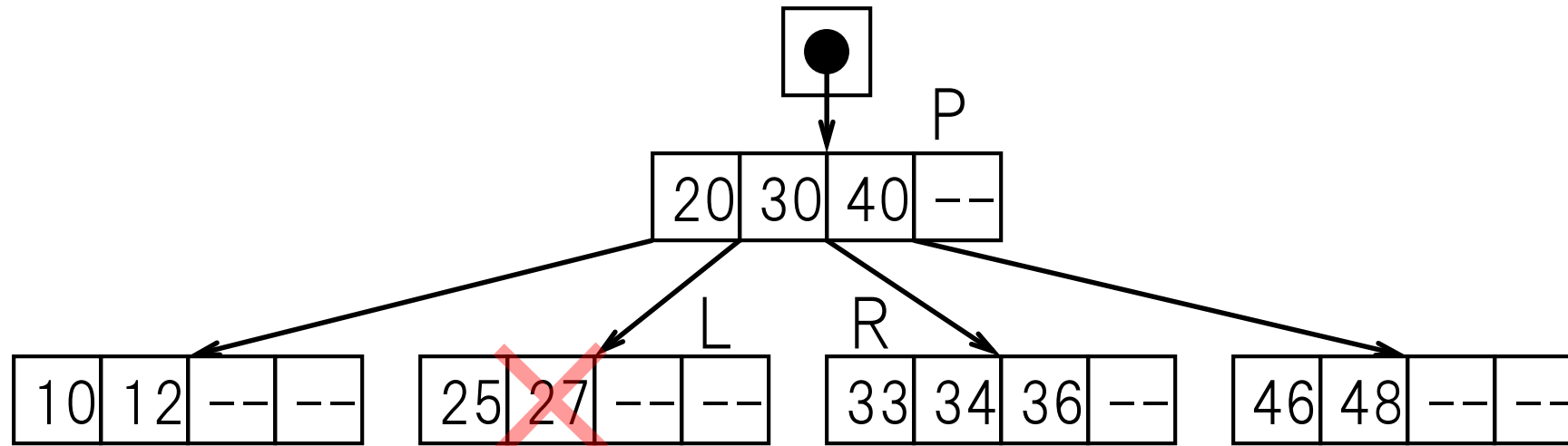
B-木の変化 : 12,18,19,4,5,6,2,3を挿入



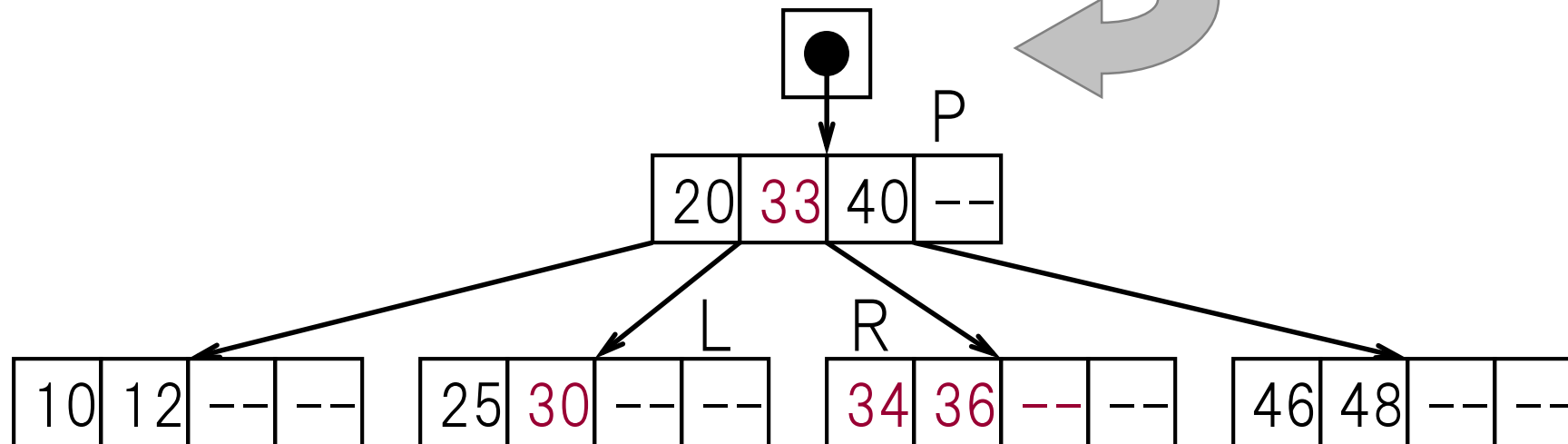
B-木の变化：1を挿入



B-木の变化：27を削除（ケース1）

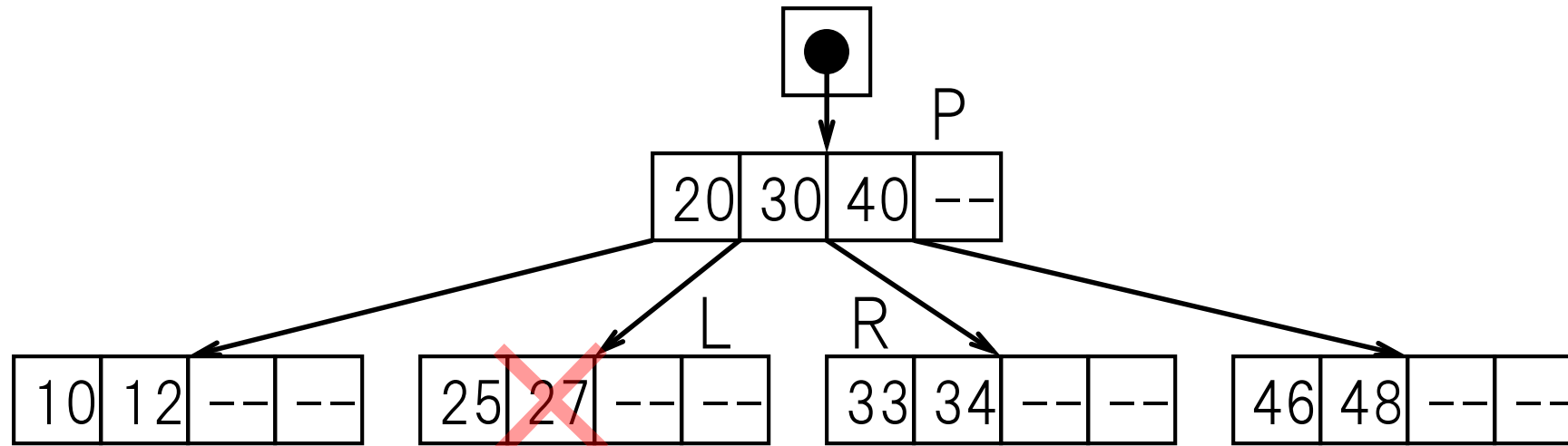


ノードLでアンダーフロー

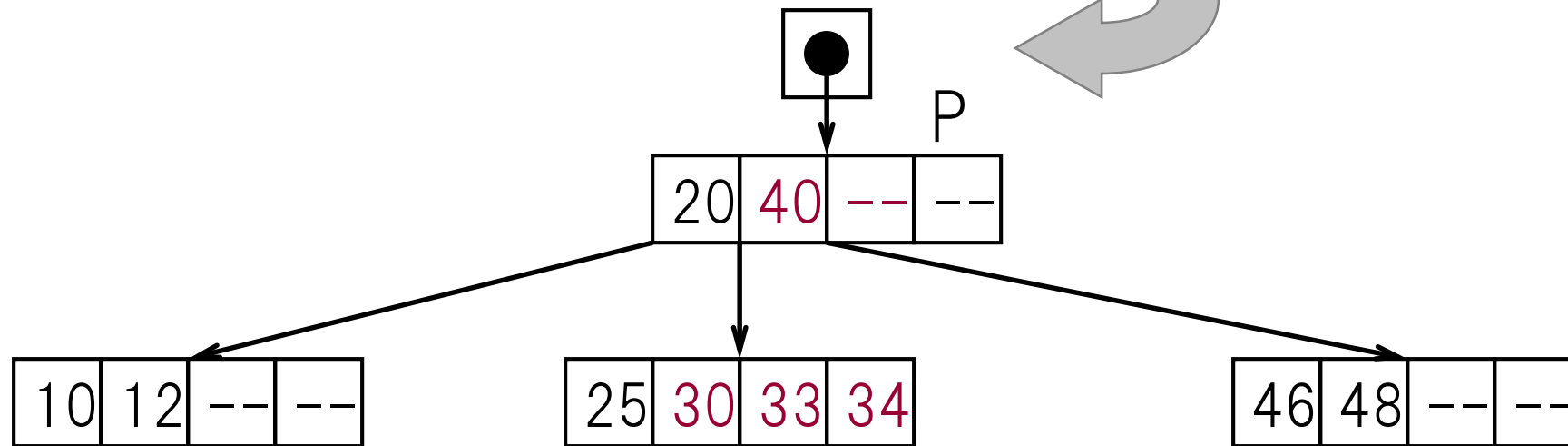


ノードRから1つ借りた

B-木の变化：27を削除（ケース2）



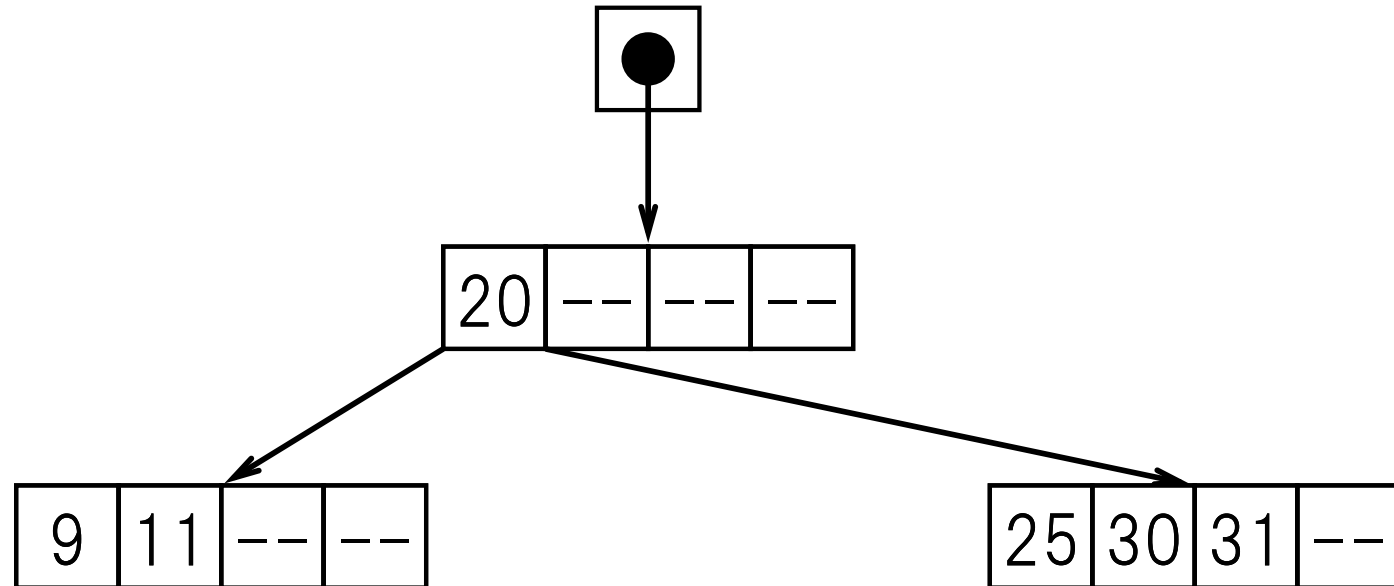
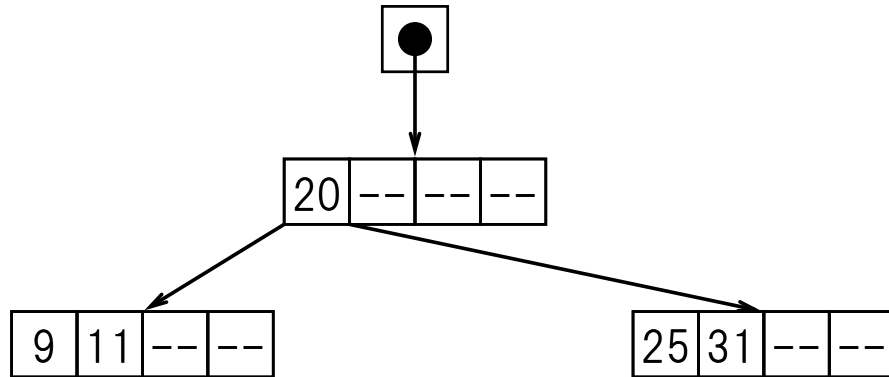
ノードLでアンダーフロー



ノードLとRを合併した

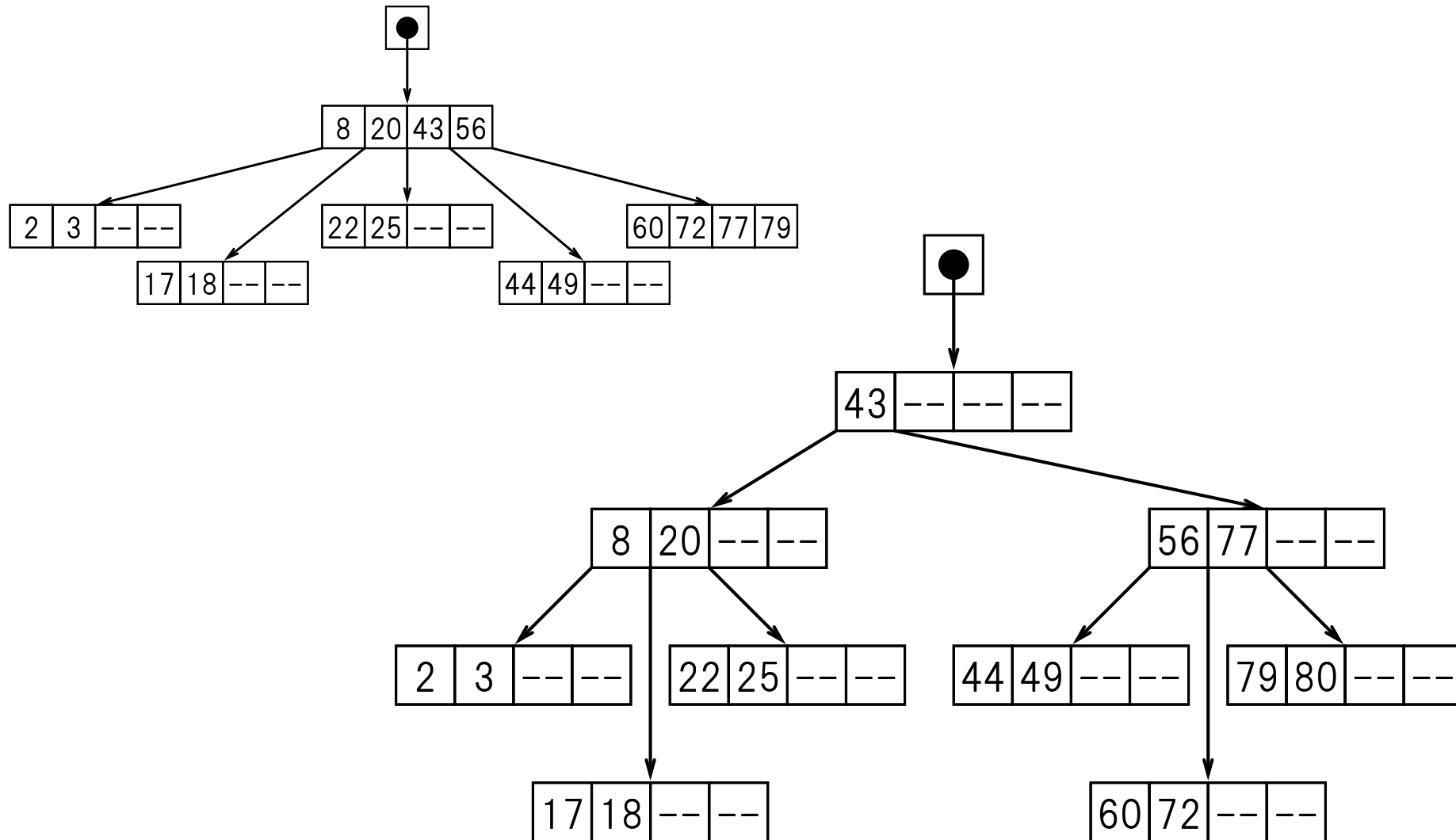
問題1

- 30 を挿入した後のB-木を答えなさい.



問題2

- 80 を挿入した後のB-木を答えなさい。



問題3

- 25 を削除した後のB-木を答えなさい.

