アルゴリズムとデータ構造
第7回 動的探索問題とデータ構造

担当: 上原隆平(uehara)

2015/04/24

動的探索問題とデータ構造

- 質問データの検索を動的な環境で行う
  i.e., データ集合に対してデータの追加と削除を許す
  - cf. 静的探索問題: データの集合が不変
- 動的探索問題の例: 役所における書類の管理
  - 転入: 住民データを台帳に追加
  - 転出: 住民データを台帳から削除
  - 住民票などの請求: 住民台帳の検索

Q. 良いデータ構造は?

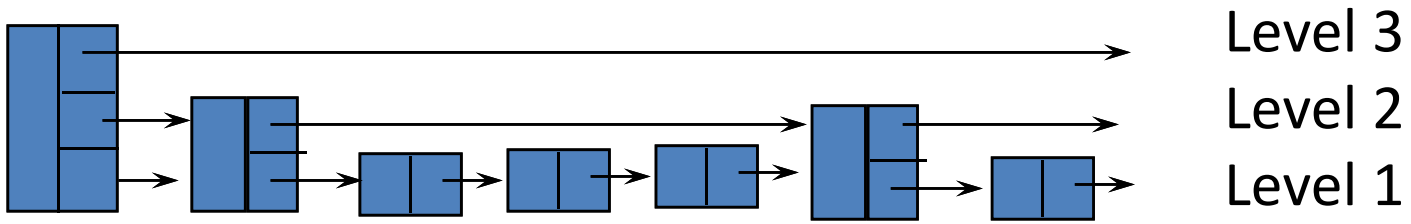
線形リストを使う

- 線形リスト: 逐次探索に対応するデータ構造

- ソート順を保つ:
  - 検索: 2分探索・O(log n)
  - 挿入・削除: 時間が掛かる
- ソート順を保たない
  - 検索・削除: 時間が掛かる
  - 挿入: 容易・O(1)

スキップリストを使う
(2重mブロック法に対応するデータ構造)

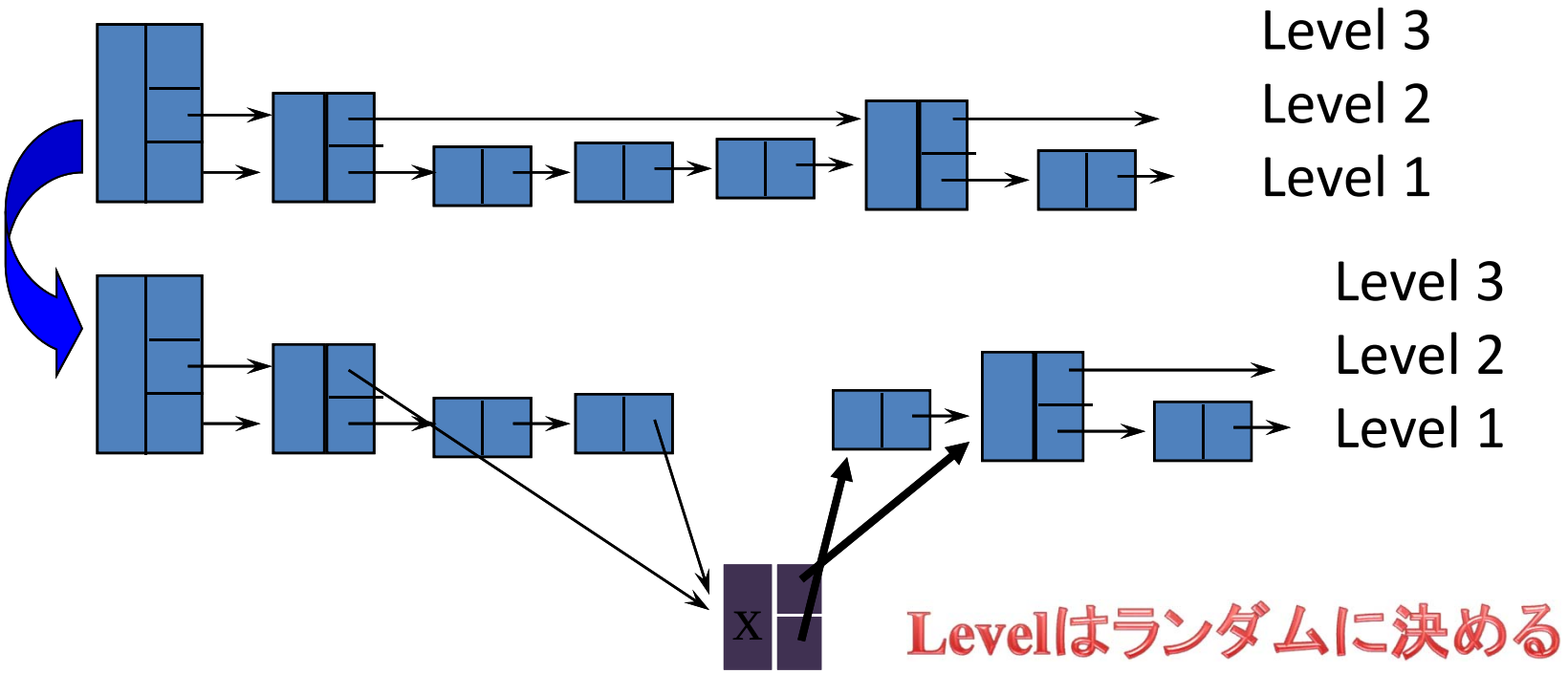
- リストのレコードは様々な「高さ」を持つ
- 高いレベルのポインタから順にポインタを辿って逐次探索の範囲を絞り込む



注意点:

- データの挿入: 新しいレコードのlevelの決定
- データの削除: 残ったレコードのlevelの変更

スキップリスト: データの挿入



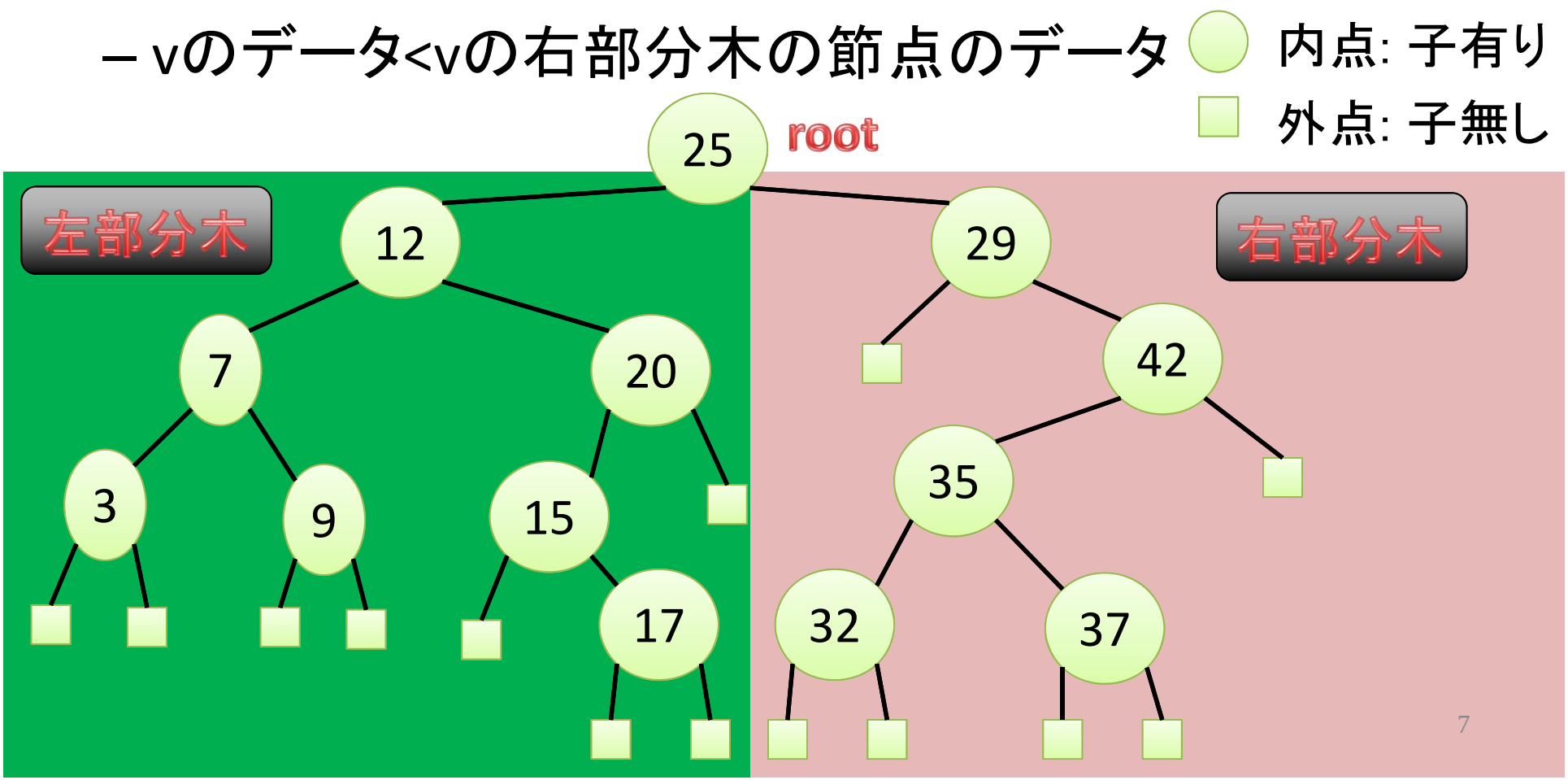
- Level 1～3のレコードの割合を保つように乱数を生成
  - e.g., 確率1/2のコインを投げて，何回目に表が出るかでlevelを決定

スキップリストの効率

- 探索時間: O(log n)
  - Level 1: n
  - Level 2: n/2
  - …
  - Level k: $n/2^k$
- 動的環境でも level k のレコードを $1/2^k$ の割合で生成すれば平均探索時間を O(log n) に保てる
  - 最悪の場合は O(log n) にならない
    →乱択アルゴリズム

2分探索法に対応するデータ構造:
動的な2分探索木 – データの蓄え方

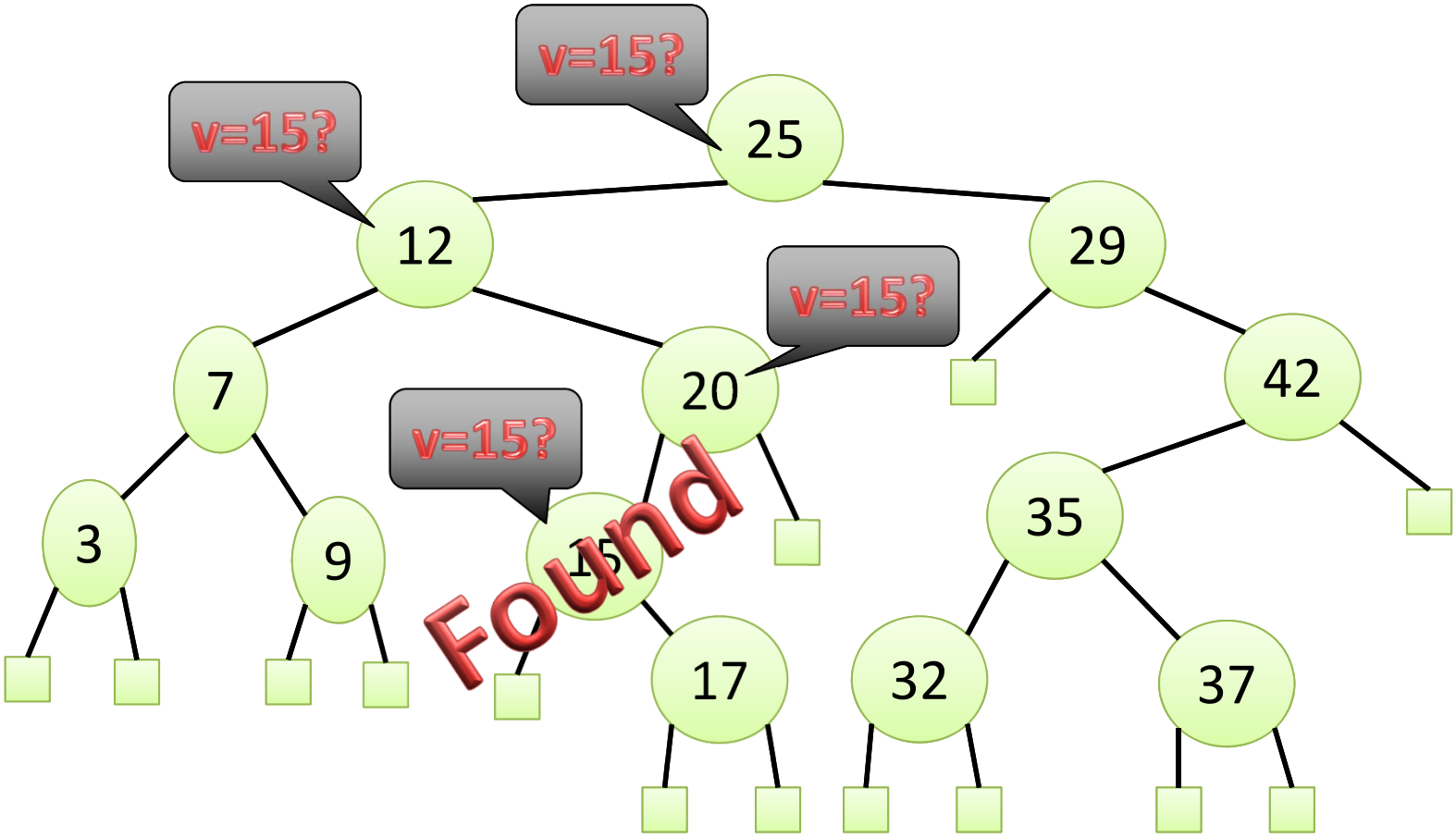
- どの節点vにおいても以下が成立
  - vのデータ>vの左部分木の節点のデータ
  - vのデータ<vの右部分木の節点のデータ



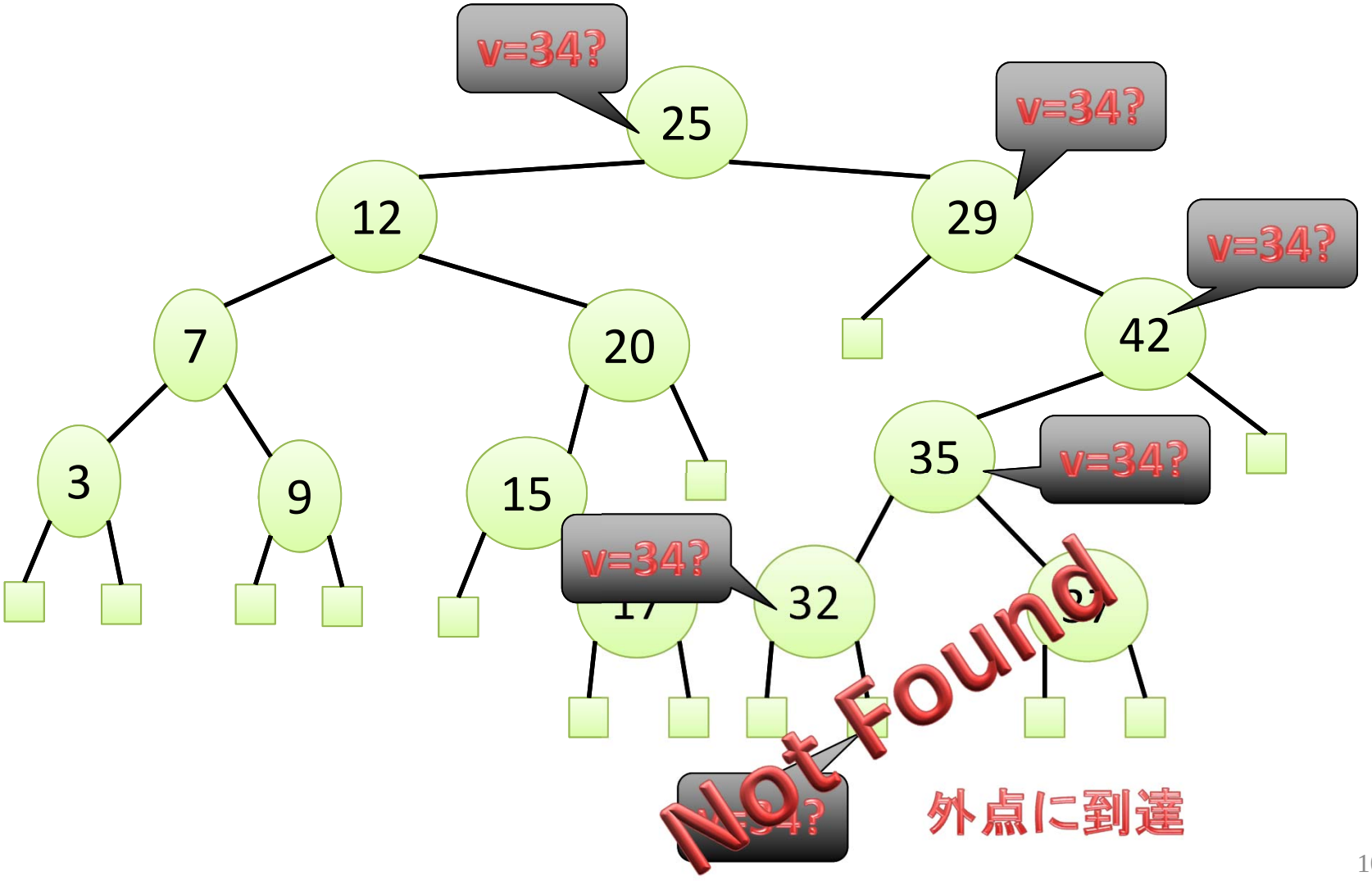
2分探索法に対応するデータ構造: 動的な2分探索木 – データの蓄え方

- 2分木の性質: どの節点vにおいても以下が成立
  - vのデータ > vの左部分木の節点のデータ
  - vのデータ < vの右部分木の節点のデータ
- 節点に多数のデータを蓄えるときは比較に用いるデータをキーとして指定しておく

2分探索木における検索: v=15の検索



2分探索木における検索: v=34の検索

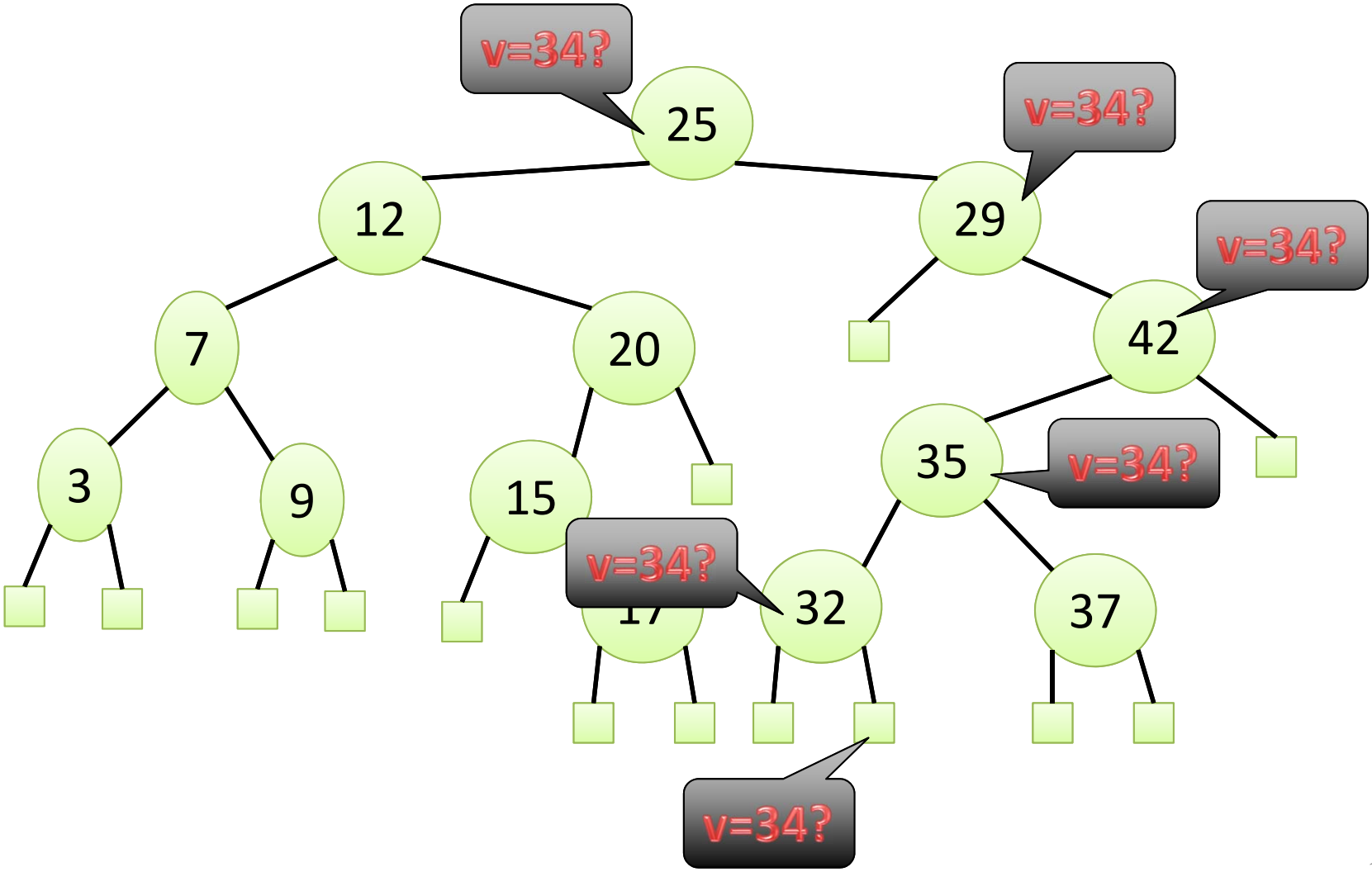


2分木へのデータの挿入･追加

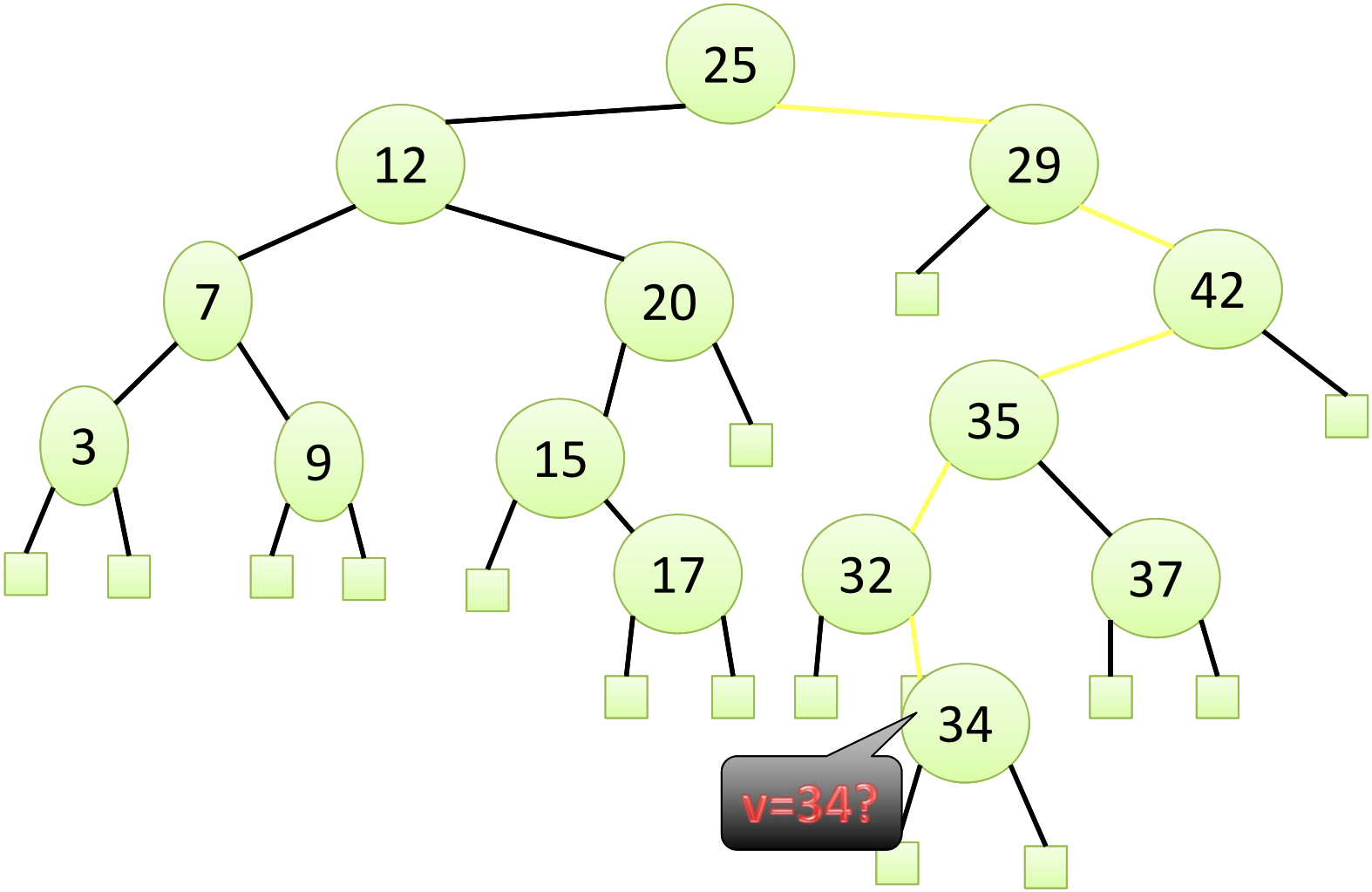
- 根(root)から2分木探索を行う
- 外点に到達したらそこにデータを蓄える

```
insert(x,tree){
  v ← root(tree);
  while( v  内点(tree) ){
     if( x <= data(v) ) then v ← vの左の子;
     else                    v ← vの右の子;
  }
  v ← 内点;
  data(v) ← x;
}
```

2分木へのデータの挿入・追加
例: x=34の追加



2分木へのデータの挿入・追加
例: x=34の追加



2分木へのデータの挿入・追加: 手続き

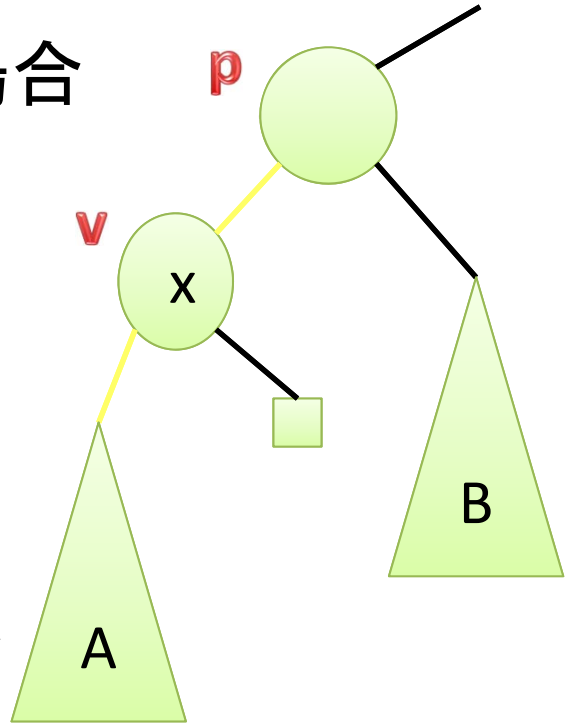
```
void insert(tree *p, int x){
  if(p == NULL){
    p = (tree*) malloc( sizeof(tree) );
    p->key = x;
    p->lchild =NULL; p->rchild = NULL;
  }else
    if( p->key < x )
      insert( p->rchild, x);
    else
      insert( p->lchild, x);
}
```

呼び出し方: insert(root,x)

rootへのポインタ

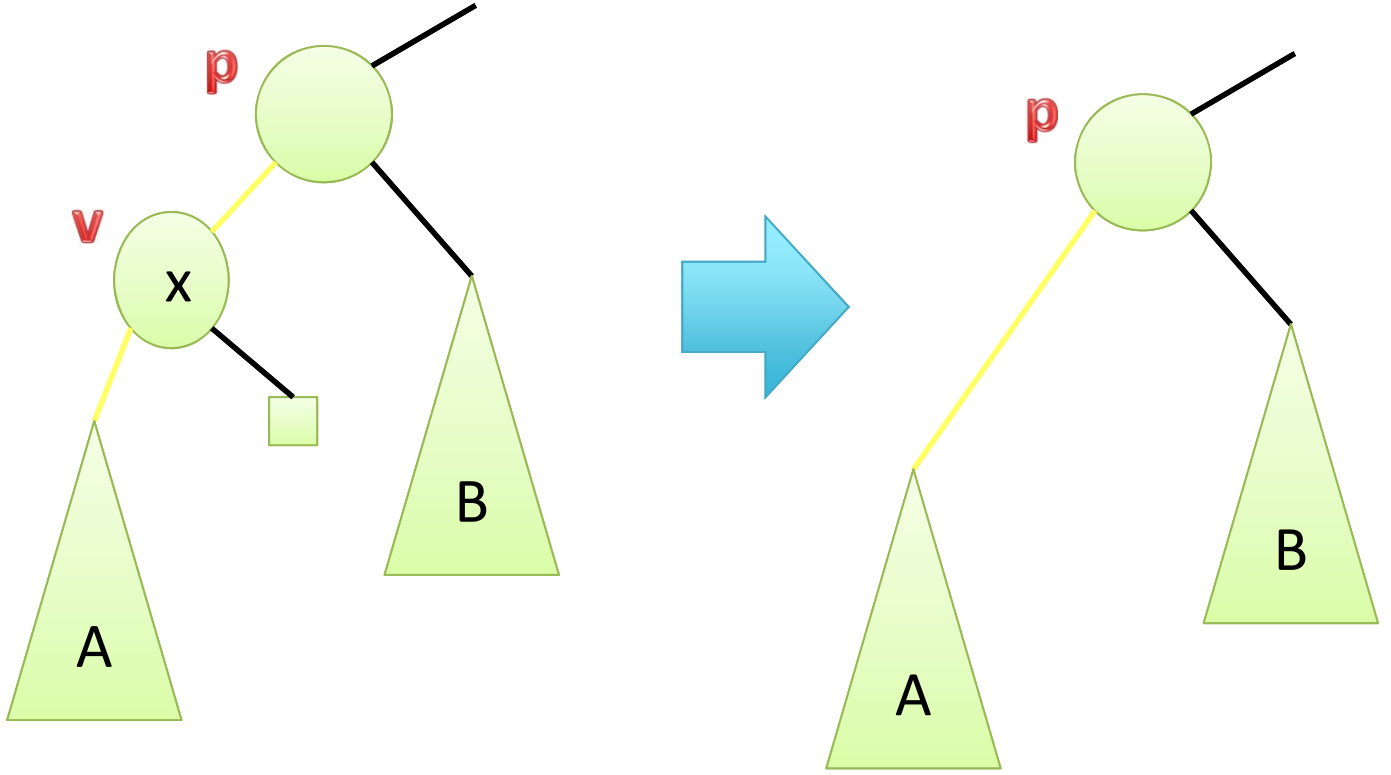
2分木からのデータの削除: キーの値がxである節点を削除

- キーxを含む節点vを見つけて場合分け
  - Case 1. vの一方の子が外点の場合
  - Case 2. vの左右の子が共に内点



2分木からのデータの削除: Case 1. vの一方の子が外点

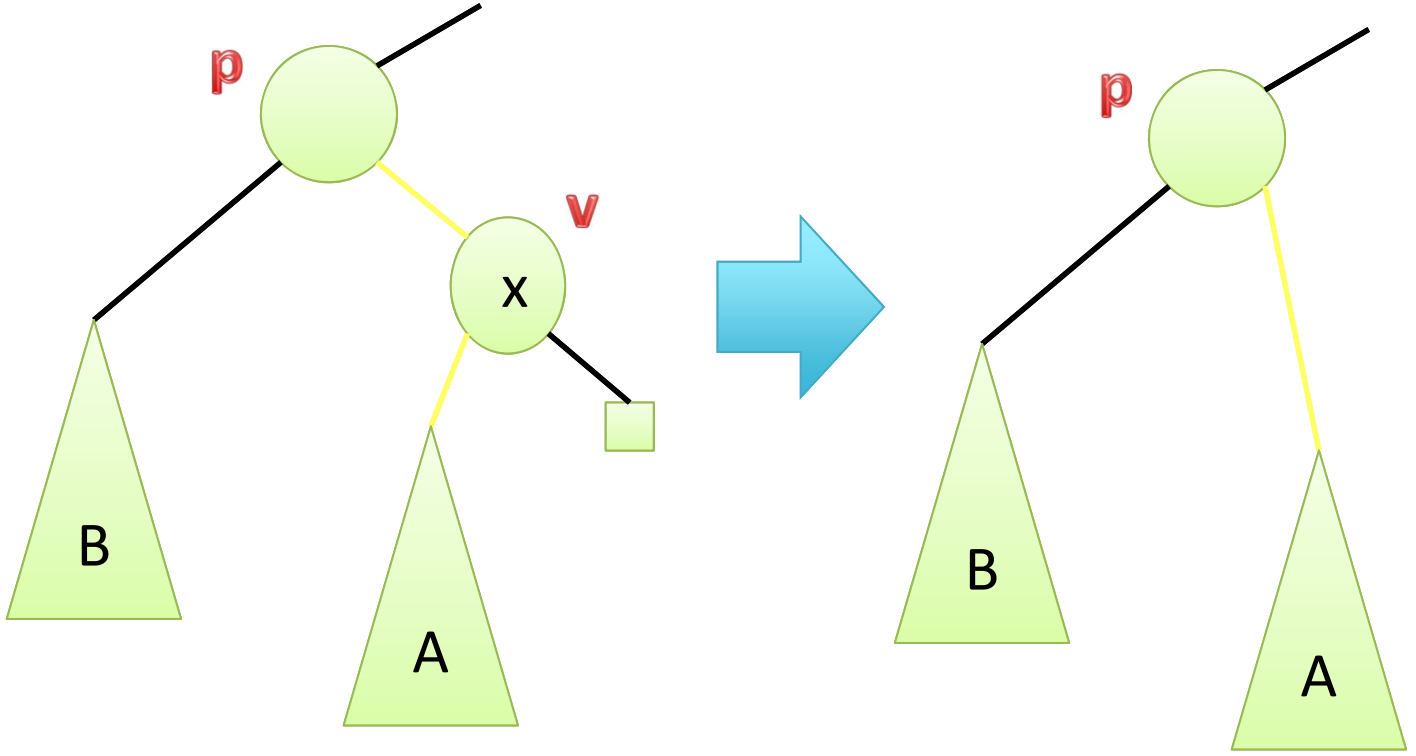
- vが親pの左の子: pの左の子=vの空でない子



Q. 2分木の性質は保たれる?

2分木からのデータの削除: Case 1. vの一方の子が外点

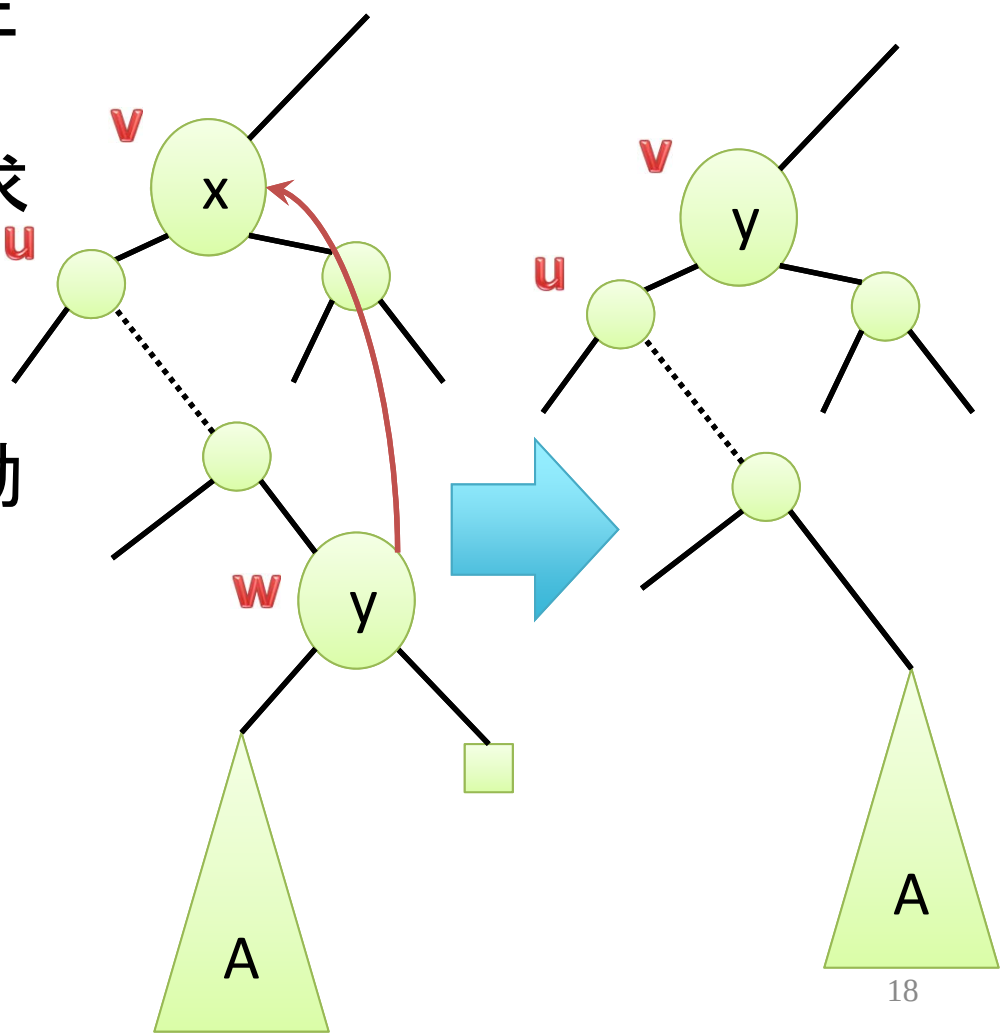
- vが親pの右の子: pの右の子=vの空でない子



2分木からのデータの削除: Case 2. vの左右の子が共に内点

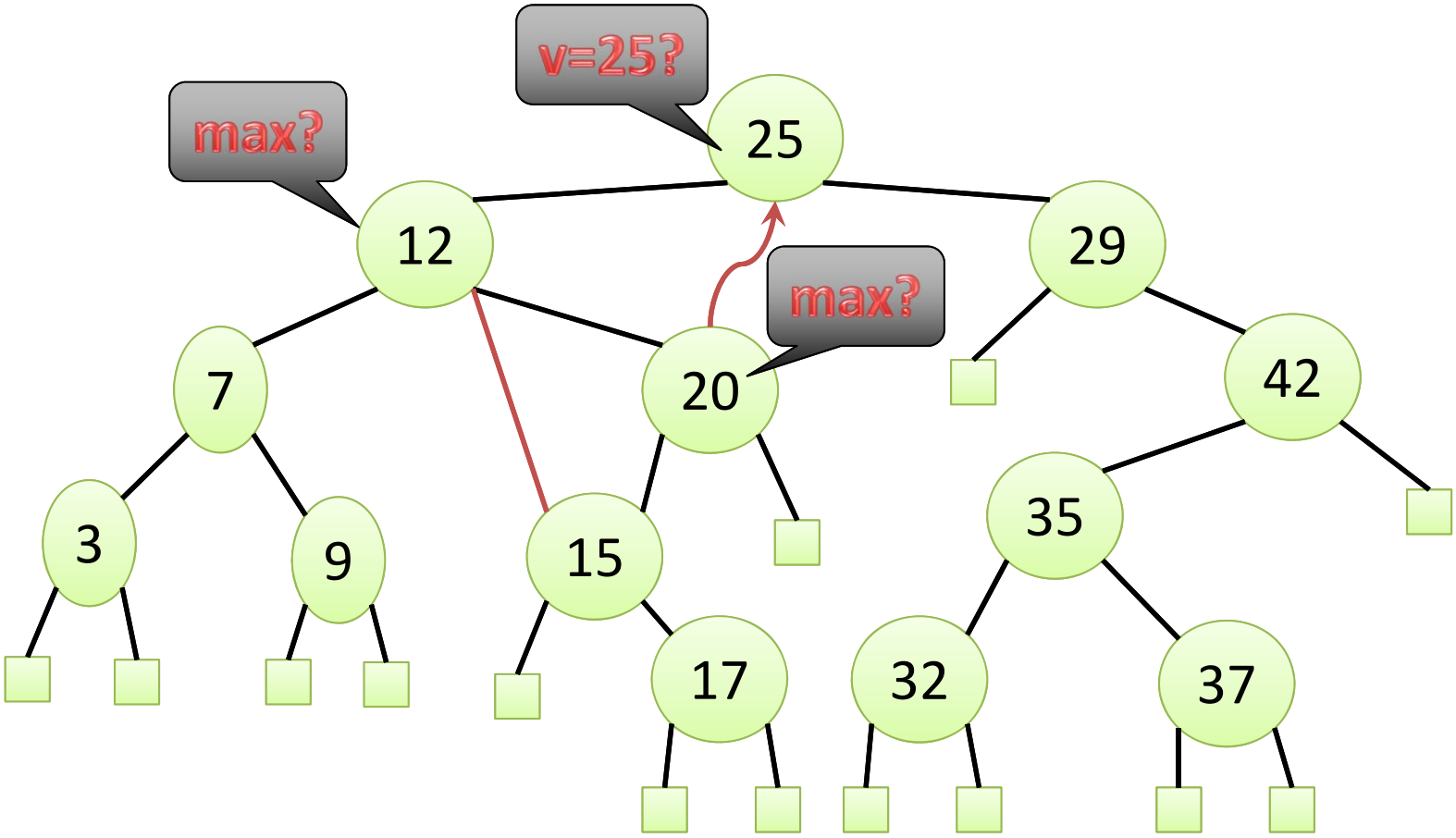
- キーxを含む節点vの左の子uの子孫で最大のキーyを含む節点wを求める
  - 節点wは右の子無し
- yの値をxの場所に移動して節点wを削除
  - Case 1に相当

Why?

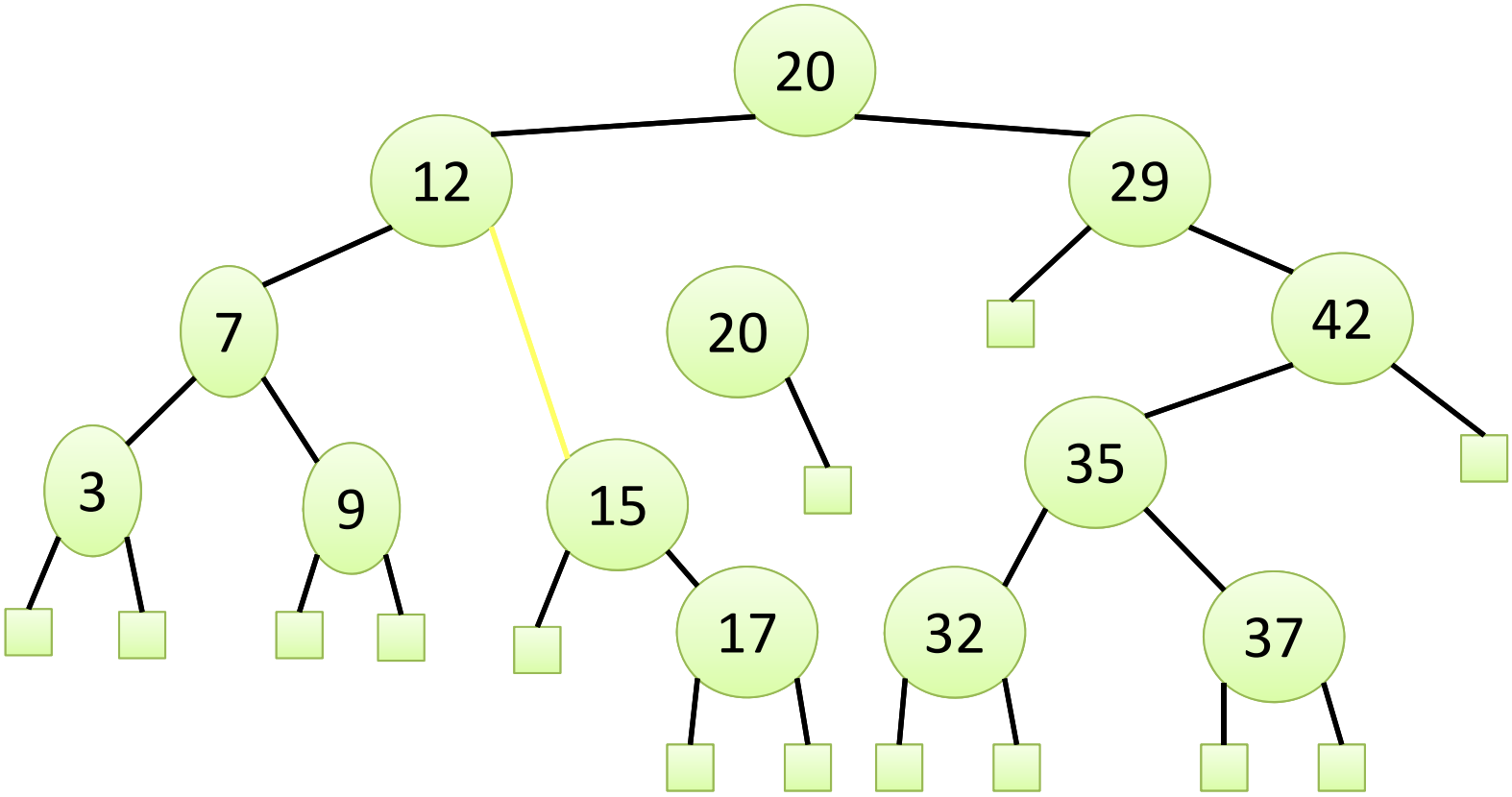


Q. 2分木の性質は?

2分木からのデータの削除:
x=25を削除



2分木からのデータの削除:
x=25を削除



5月1日は休講

ミニ演習

5月8日はレポート締切

- 前回分: データnの2分ヒープの深さを求めよ
- 以下の一連の操作の結果をそれぞれ示せ
  - (1) 12を削除，(2) 7を削除，(3) 12を挿入