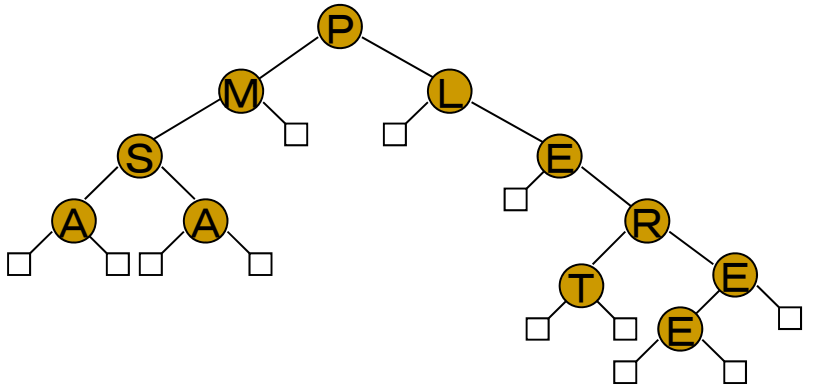
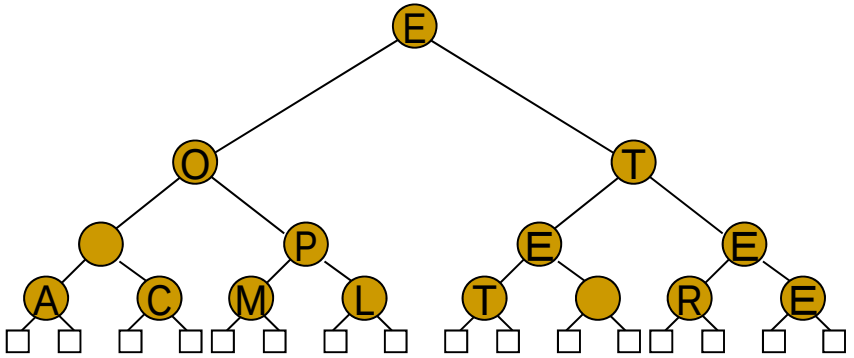
アルゴリズムとデータ構造

- 第7回講義トピック：木（続き）
  - 木の性質
  - 応用：2分探索木
  - その他
    - 一般的な木から2分木への変換
    - 配列による木および森の表現

木の性質

- 木において、任意の2つの節点を結ぶパスはただ1つである。
- 節点の個数がNである木には、辺がN - 1個ある。
- 内部節点がNである2分木には、N+1個の外部節点がある。
- 内部節点がNである2分木が一杯になっていれば、その高さは約log2Nである。





2分探索木 (binary search tree)

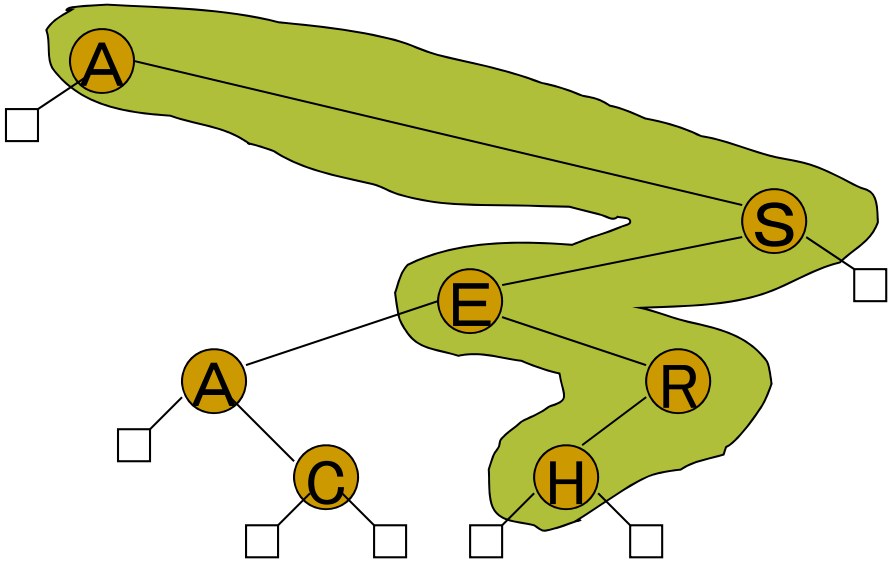
- 探索とは前もって格納されている多くのデータの中から望みのデータを探し出す操作をいう。与えられた探索キーと一致するキーをもつレコードを見つけ出し、必要なレコード内のデータを引き出す。
- 2分探索木
  - 2分探索木はどのサブツリーをとっても、ルートは左の要素よりも大きく、右の要素よりも小さいという関係が成り立つ。
  - 2分探索木における探索は与えられたキーをルートにあるキーと比較し、小さければ左部分木を探索し、等しい時は停止し、大きい時には右部分木を探索する。以上のプロセスを再帰的に適用する。

探索のプログラム例

```
static struct node {
  int key, info;
  struct node *l, *r;
};
static struct node *t, *head, *z;
int treesearch(int v) {
    struct node *x = head->r;
    z->key = v;
    while (v != x->key)
            x = (v < x->key) ? x->l : x->r;
    return x->info;
}
```

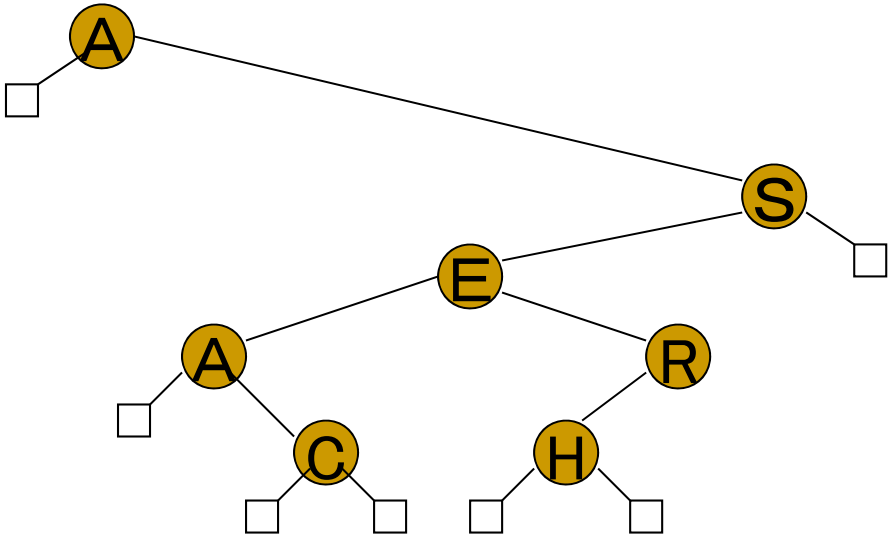
探索の過程

- 以下の2分探索木において、節点Iを探索する時の様子を示す。この探索では、節点Iは存在しないので、探索は不成功に終わる。



2分探索木の挿入

- 空の2分探索木において、ASEARCHINGEXAMPLEの順に要素を一部（下線部分）挿入した時の例を示す。



2分探索木の初期化と中央順走査プログラム

- 2分探索木の初期化

```
treeinitialize() {
  z = (struct node *) malloc(sizeof *z);
  z->l = z; z->r = z; z->info = -1;
  head = (struct node *) malloc(sizeof *head);
  head->r = z; head->key = 0;
}
```

- 2分探索木の中央順走査

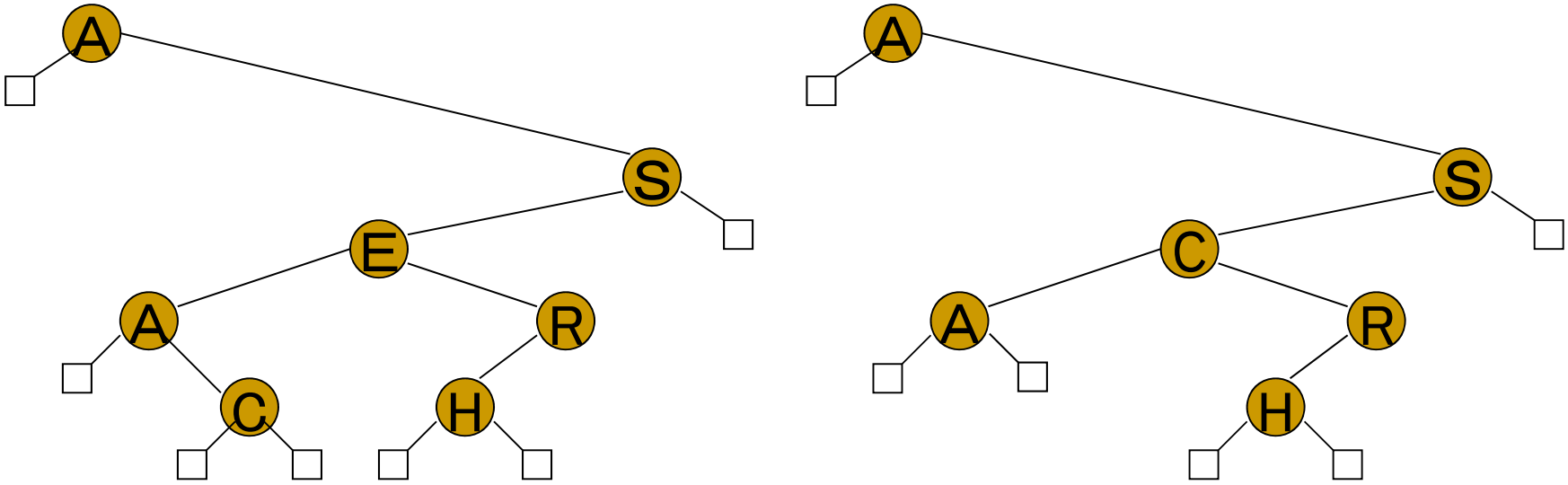
```
treeprint() { treeprintr(head->r); }
treeprintr(struct node *x) {
  if (x != z) {
    treeprintr(x->l); printnode(x); treeprintr(x->r);
  }
}
```

2分探索木のにおける要素の挿入

```
treeinsert(int v, int info) {
  struct node *p, *x;
  p = head; x = head->r;
  while (x != z) {
    p = x;
    x = (v < x->key) ? x->l : x->r;
  }
  x = (struct node *) malloc(sizeof *x);
  x->key = v;
  x->info = info;
  x->l = z;
  x->r = z;
  if (v < p->key) p->l = x;
  else p->r =x;
}
```

2分探索木における要素の削除

- 3つの場合に分けて考える
  - 子を持たない節点はそのまま削除する。
  - 子を1つもつ節点は子節点で置き替える。
  - 子を2つもつ節点は左部分木の最大節点かあるいは右部分木の最小節点で置き替える。以下は節点Eを削除する例である。

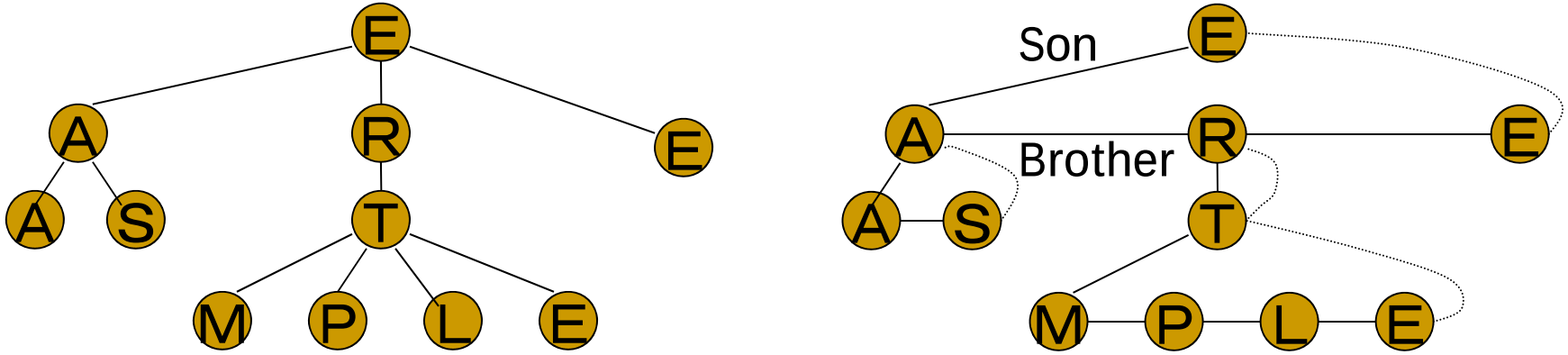


2分探索木の計算量

- N個のキーからなる2分探索木での探索
  - 最悪の場合にはN回の比較が必要になる。
  - ランダムなキーから生成された場合は平均約 2lnN 回の比較が必要になる。

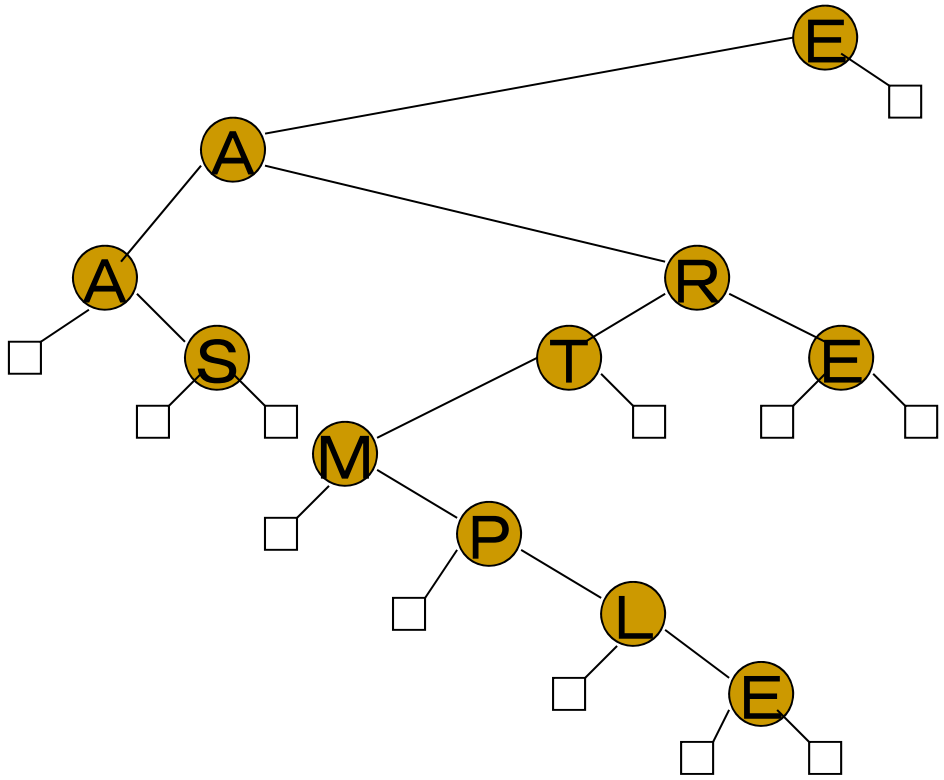
一般的な木から2分木への変換

- 一般的な木は2以上の子を持つので、木の表現が複雑になる。
- 解決法の1つとして各節点のリンクを2つとし、それぞれ左端の子と右の兄弟を指すようにする。



一般的な木から2分木への変換

- このように変形した後の木は2分木そのものである



配列による木あるいは森の表現

- 子から親へのリンクだけを考える場合

k	1	2	3	4	5	6	7	8	9	10	11
a[k]	A	S	A	M	P	L	E	T	R	E	E
dad[k]	3	3	10	8	8	8	8	9	10	10	10