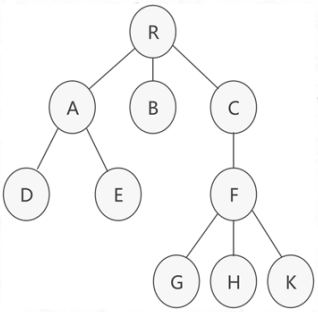
树的三种表示法

如何存储具有普通树结构的数据:



1.双亲表示法:

双亲表示法采用顺序表（也就是数组）存储普通树，其实现的核心思想是：顺序存储各个节点的同时，给各节点附加一个记录其父节点位置的变量。

注意，根节点没有父节点（父节点又称为双亲节点），因此根节点记录父节点位置的变量通常置为 -1。

采用双亲表示法存储上图中普通树，其存储状态如下所示

数组下标	data	parent
0	R	-1
1	A	0
2	B	0
3	C	0
4	D	1
5	E	1
6	F	3
7	G	6
8	H	6
9	K	6

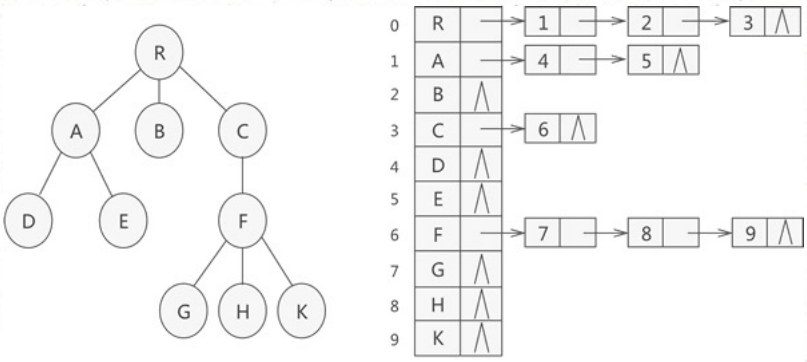
```
class Node {
  constructor(data) {
    this.data = data;
    this.parent = null;
  }
}
function set_nodes(obj) {
  let node, res = [];
  for (let key in obj) {
     node = new Node(key);
     node.parent = obj[key];
     res.push(node);
   }
   return res;
```

```
}

function findParent(n) {
  let parent_index;
  for (let i = 0; i < res.length; i++) {
    if (res[i].data === n) {
      parent_index = res[i].parent;
      console.log(res[parent_index]);
      continue;
    }
  }
  return
}
const letters = { 'A': 0, 'B': 0, 'C': 0, 'D': 1, 'E': 1, 'F': 3, 'G': 6, 'H': 6, 'K':
const res = set_nodes(letters);
findParent("G")
```

2.孩子表示法:

孩子表示法存储普通树采用的是 "顺序表+链表" 的组合结构，其存储过程是：从树的根节点开始，使用顺序表依次存储树中各个节点，需要注意的是，与双亲表示法不同，孩子表示法会给各个节点配备一个链表，用于存储各节点的孩子节点位于顺序表中的位置。如果节点没有孩子节点（叶子节点），则该节点的链表为空链表。



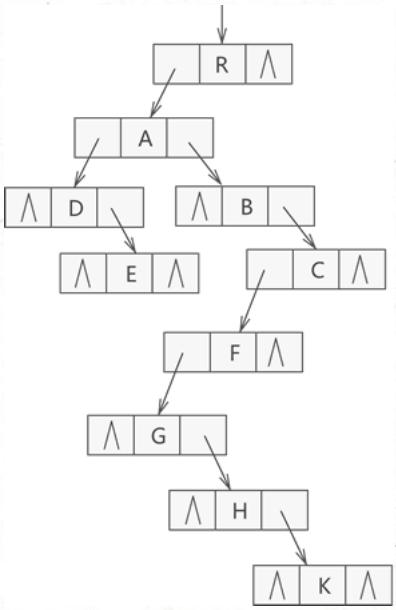
a) 普通树　　b) 孩子表示法

使用孩子表示法存储的树结构，正好和双亲表示法相反，适用于查找某结点的孩子结点，不适用于查找其父结点。

3.孩子兄弟表示法:

孩子兄弟表示法，采用的是链式存储结构，其存储树的实现思想是：从树的根节点开始，依次用链表存储各个节点的孩子节点和兄弟节点。

1. 节点的值；
2. 指向孩子节点的指针；
3. 指向兄弟节点的指针；



节点 R 无兄弟节点，其孩子节点是 A；节点 A 的兄弟节点分别是 B 和 C，其孩子节点为 D，依次类推