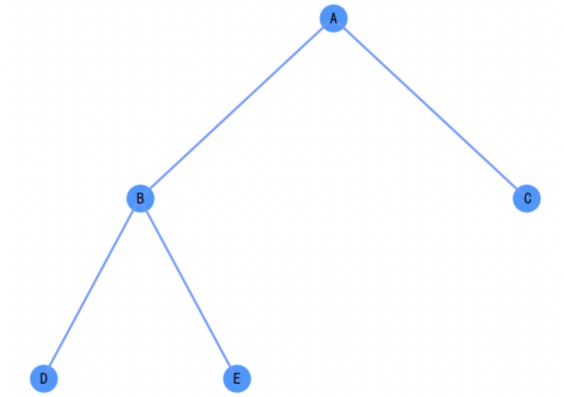
选20、二叉树的存储与表示

一、二叉树的数组存储方案

1）用数组来存储二叉树，如果是完全二叉树，就可以用数组按层来存储：

tree=['A','B','C','D','E']

```
import DrawBinTree as dbt
dbt.draw_full_tree(["A",'B','C','D','E'])
```



对于某数组中某元素，设该元素的位置为index，则有：

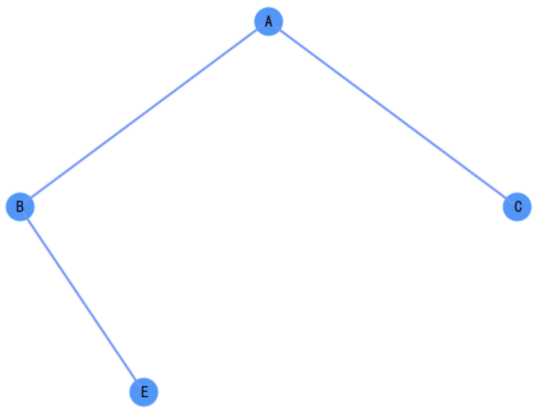
层level=int(math.log2(index))+1

从左往右第n个，n=index-2**(level-1)-1

2）如果是非完全二叉树，也可以用数组按层来存储：

tree=['A','B','C','','E']

```
dbt.draw_full_tree(['A','B','C','','E'])
```



但是可以看到存在存储空间的浪费

二、用链表来存储二叉树

二叉树的某一个节点包含三条信息：

1）节点内容

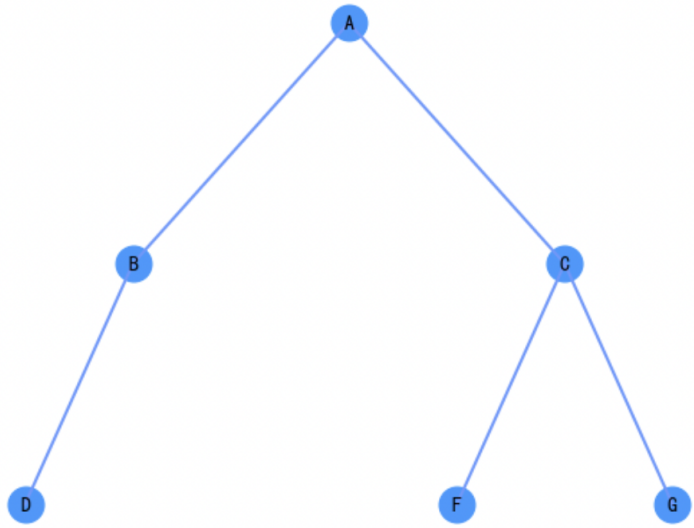
2）左孩子节点指针

3）右孩子节点指针

于是一个节点node是三元组，如：

node_tree=[['A',1,2],['B',3,-1],['C',4,5],['D',-1,-1],['F',-1,-1],['G',-1,-1]]

```
dbt.draw_link_tree([['A',1,2],['B',3,-1],['C',4,5],['D',-1,-1],['F',-1,-1],['G',-1,-1]])
```



注意，节点指针为-1表示没有后继节点

三、用Python可列举对象嵌套来实现二叉树存储

如：

tree=['A',['B',['C',None,None],None],['D',['F',None,None],None]]

```
dbt.draw_list_tree(listtree=['A',['B',['C',None,None],None],['D',['F',None,None],None]])
```