

太戈编程
etiger.vip

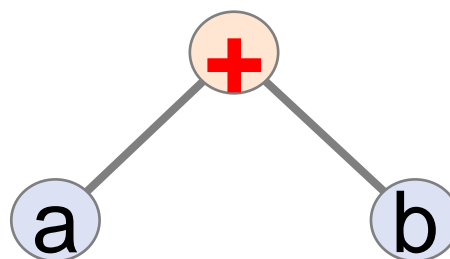
数据结构

表达式树

表达式

$a+b$

表达式树



后序遍历得到
后缀表达式

$ab+$

前序遍历得到
前缀表达式

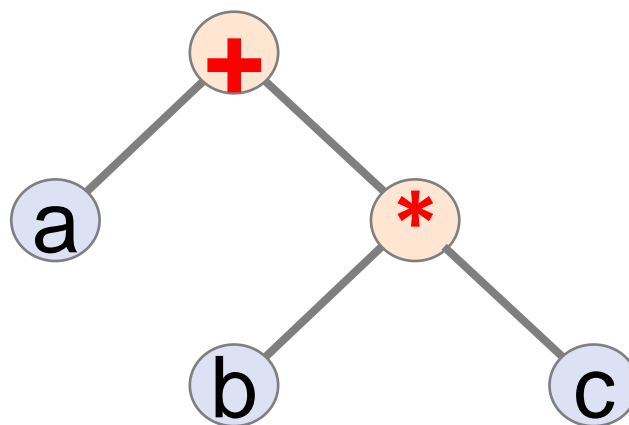
$+ab$

最后运算
作为根

表达式

$a + b * c$

表达式树



后序遍历得到
后缀表达式

$abc * +$

前序遍历得到
前缀表达式

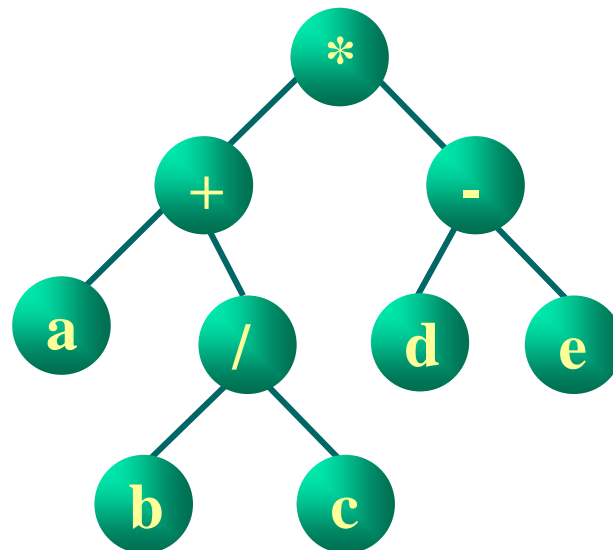
$+ a * bc$

最后运算
作为根

表达式树

表达式可以转为二叉树
叶子节点表示运算数
分支节点表示运算符

$(a+b/c)*(d-e)$ 转为二叉树



后缀表达式
 $a b c / + d e - *$

前缀表达式
 $* + a / b c - d e$

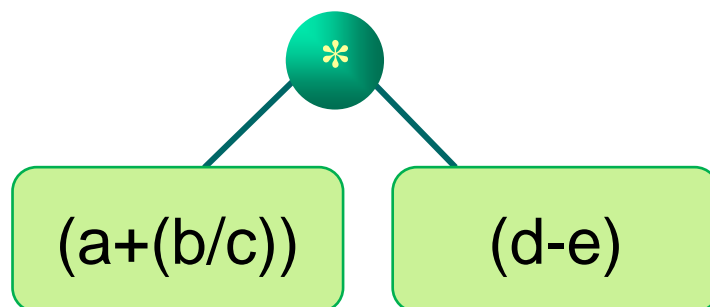
最后运算
作为根

构建表达式树

第一步：找到最后的运算符

第二步：最后运算符作为根节点，运算符左右两侧的表达式作为根节点的左右子树

表达式 $(a+b/c)*(d-e)$



最后运算
作为根

表达式树

写出表达式 $a+b*(c-d)-e/f$ 的前缀表达式和后缀表达式

画出表达式树

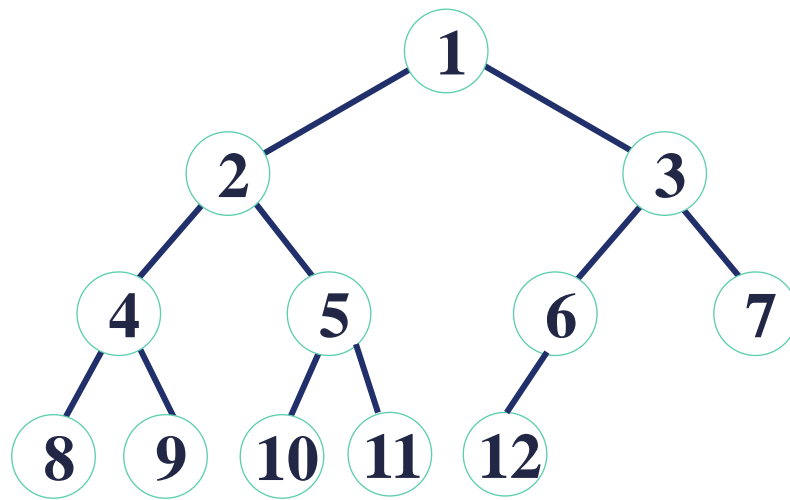
前缀表达式

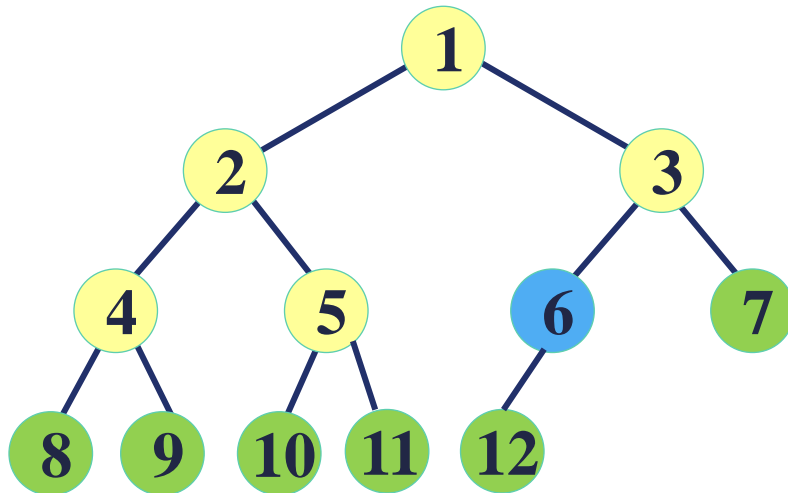
$- + a * b - c d / e f$

后缀表达式

$a b c d - * + e f / -$

1366.数儿子

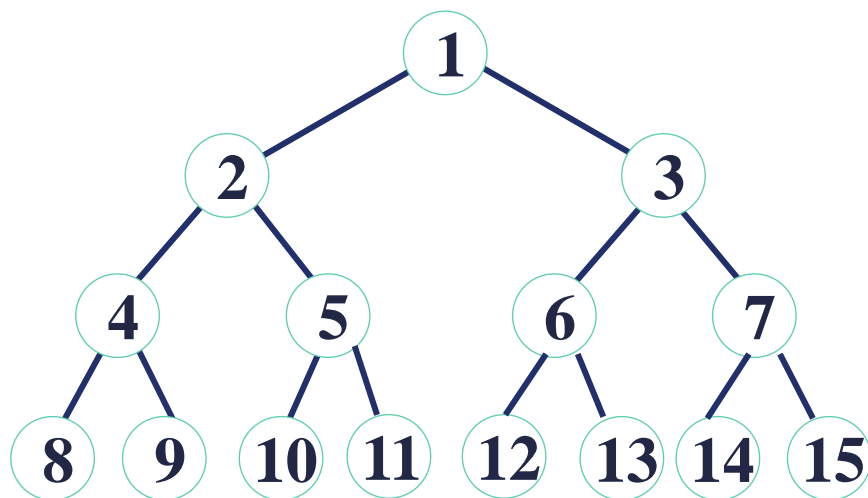




请同学简述题意
突出核心要点

完全二叉树有 n 个节点，请问：
有多少节点是叶节点（没有儿子）？
有多少节点拥有1个儿子节点？
有多少节点拥有2个儿子节点？

完全二叉树



25号结点在第几层?

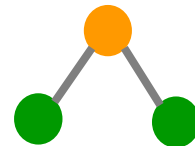
52号结点在第几层?

70号结点在第几层?

重大发现：
拥有1个儿子节点数量为1或0

结点数	层数
1	1
2	2
3	2
4	3
5	3
6	3
7	3
8	4
9	4
10	4
11	4
12	4
13	4
14	4
15	4

任意二叉树满足 021 定理

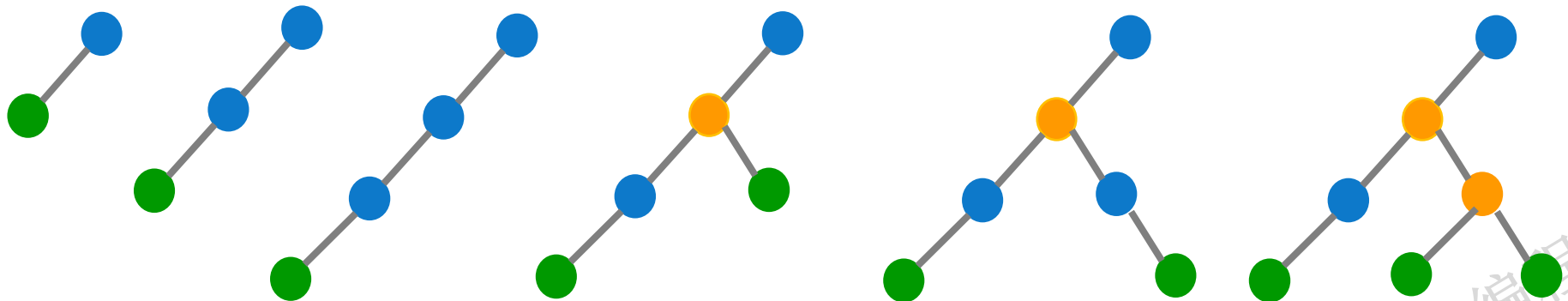


$$n_0 = n_2 + 1$$

有0个儿子的
结点数为 n_0

有2个儿子的
结点数为 n_2

(叶子数)



```
6   int n,ans0,ans1,ans2;  
7   cin>>n;  
8   if(n%2==0) ans1=1;  
9   else   
10  ans2=(n-ans1)/2;  
11  ans0=  
12  cout<<<<endl;
```

n是偶数时有1个节点
拥有正好1个儿子

n是奇数时没有节点
拥有正好1个儿子

为什么？

栈

stack

先进后出

FILO

first-in-last-out

请预测
输出结果

```
1 #include<iostream>
2 #include<stack>
3 using namespace std;
4 int main(){
5     stack<int> s;
6     s.push(3);
7     s.push(1);
8     s.push(4);
9     cout<<s.size()<<endl;
10    cout<<s.top()<<endl;
11    s.pop();
12    cout<<s.size()<<endl;
13    cout<<s.top()<<endl;
14    s.pop();
15    cout<<s.size()<<endl;
16    cout<<s.top()<<endl;
17    s.pop();
18    cout<<s.size()<<endl;
19    return 0;
20 }
```

3		
3	1	
3	1	4

3	1
---	---

3

2084.口算王

请同学简述题意
突出核心要点

给定一条计算式子只有加号+
减号-乘号*，求计算结果

用具体样例体会算法步骤

$2+3$

2先储存

$2+3*4$

2先储存, 3和4相乘

$3-1-1$

3和1先相减

$9-2*2-1$

9先储存, 2和2相乘

请留意各个运算的先后顺序

第一个数字是否第一个计算?

表达式计算：双栈协作

动态 进栈 出栈	2 个 栈
----------------	-------------

数字栈

符号栈

表达式计算：双栈协作


动态 进栈 出栈	2 个 栈
----------------	-------------

2 3 4	数字栈
+ *	符号栈

$2+3*4$

```
3  stack<int> num;  
4  stack<char> op;
```

数字栈+符号栈

```
21  int a,b;  
22  char c;  
23  cin>>a;  
24  num.push(a);  
25  while(cin>>c>>b){  
26      if(c=='*') {  
27          a=num.top();  
28          num.pop();  
29          a*=b;  
30          num.push(a);  
31      }  
32      else{  
33          solve();  
34            
35      }  
36  }  
37  }  
38  solve();  
39  cout<<num.top()<<endl;
```

遇到乘号*
直接计算

solve()
将栈内
剩余运算完成

solve()
将栈内
剩余运算完成

```
5 void solve(){  
6     while(!op.empty()){  
7         int z=op.top();  
8         op.pop();  
9         int y=num.top();  
10        num.pop();  
11        int x=num.top();  
12        num.pop();  
13        if(z=='+')  
14        else  
15        num.push(x);  
16    }  
17 }
```

两种情况
加号或减号

太戈编程

1366.数儿子

2084.口算王