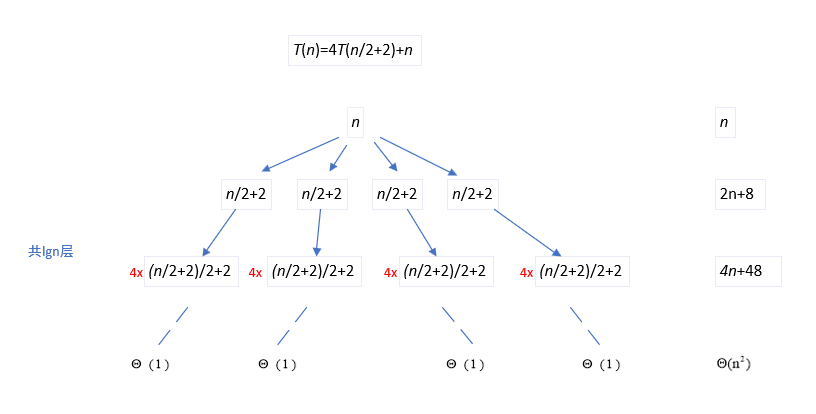
1. **作业一**

对递归式*T*(*n*)=4*T*(*n*/2+2)+*n*，利用递归树确定一个好的渐近上界，用代入法进行验证。

解答：



确定渐进上界：

代入法验证：



当

1. **作业二**

对下列每个递归式，给出*T*(*n*)的渐近上界和下界。假定*n*2时T(*n*)为常数。给出尽量紧确定的界，并验证其正确性。  
a. *T*(*n*)=2*T*(*n*/2)+*n*4b. *T*(*n*)=7*T*(*n*/10)+*n*c. *T*(*n*)=16*T*(*n*/4)+*n*2d. *T*(*n*)=7*T*(*n*/3)+*n*2e. *T*(*n*)=7*T*(*n*/2)+*n*2  
f. *T*(*n*)=2*T*(*n*/4)+g. *T*(*n*)=*T*(*n-*2)+*n*2

解答：

a. 主方法



b.主方法

c.主方法



d.主方法



e.主方法



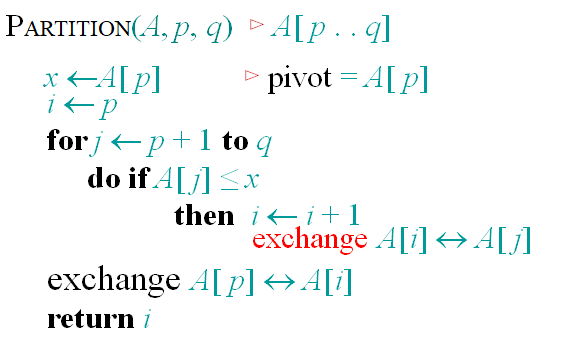
f.主方法



g.设d=n mod 2



1. **作业三**



请用循环不变式证明快速排序算法中Partition过程的正确性

解答：

代码第3-6行，对任意数组下标k，循环不等式：

证明：

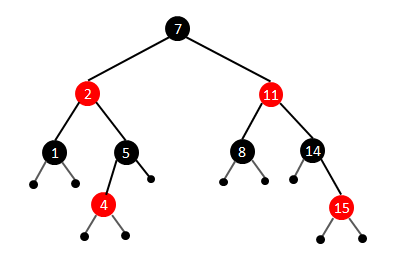
**初始化：**在循环的第一轮迭代开始之前，i=p，j=p+1，。而i+1和j-1之间没有数值，所以前两个条件满足。第一行的赋值操作使得第3个条件满足。

**保持：**主要分两种情况：（1）A[j]>x时，循环就执行了j=j+1，对A[j-1],条件2成立，其他项都保持不变。（2）A[j]<=x时，i=i+1，交换了A[i]和A[j]，再进行j=j+1。因为进行了交换，现有A[i]<=x,条件1满足。类似也能得到A[j-1]>x，条件2满足。因为根据循环不变量，被交换到A[j-1]的值总是比x大。

**终止：**当终止时，j=q+1,数组中的每个元素必然属于循环不变量所描述的三个集合的一个。也就是说，已经将数组中的所有元素划分成了三个集合: 。

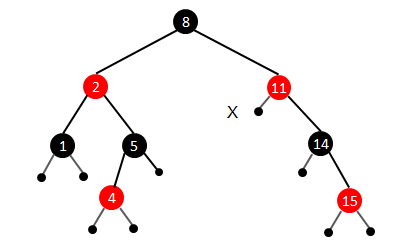
1. **作业四**

题目：删除红黑树中的节点7



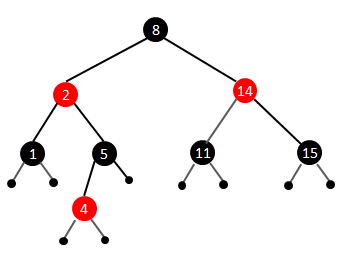
**解答：**

（1）将节点7的键值与其后继节点8进行交换，并删除新的7号节点，如下图



（2）调整：X的兄弟节点14是黑色的，同时兄弟节点14的右孩子节点15为红色。故

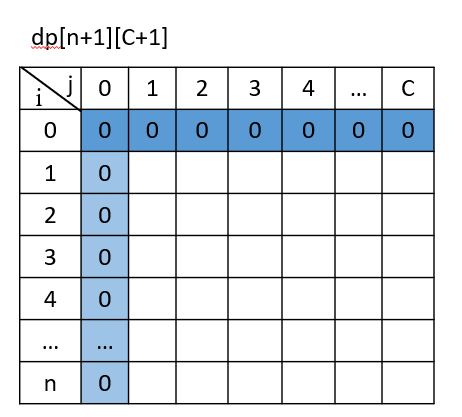
* 将X父节点11颜色赋值给X的兄弟节点14。
* 将X父节点11设为“黑色”。
* 将X兄弟节点14的右子节点15设为“黑色”。
* 对X的父节点11进行左旋。



**5.作业五：0-1背包问题求解**

有一个背包，容量为C ( Capacity )，现在有n种不同的物品编号分别为1...n。其中每一件物品的重量为w(i) ，价值为v(i)，请设计求解算法使得在不超过背包容量的前提下背包内物品价值最大，并分析所设计算法的时间性能。

**（1）解析：**设置一个数组dp[n+1][C+1]，其中dp[i][j]用于存放在背包容量不超过j的情况下，前i个物品最佳组合对应的价值。



对于dp[i][j]，分为两种情况，第一种就是w[i]>j(即背包不能容下第i个物品)，其值就为dp[i-1][j];第二种就是w[i]<=j，有两种选择，选择第i号物品进入背包，也可以不选择第i号物品进入背包，需要取两者中最优的值赋给dp[i][j]。可得其中递推关系公式为：

****

**（2）伪代码：**

**Backpack0\_1(dp,n,C)**

**1. for j ← 0 to C //第一行初始化为0**

**2. dp[0] [j]← 0；**

**3. for j ← 1 to n //第一列初始化为0**

**4. dp[j] [0]← 0；**

**5. for i ← 1 to n**

**6. for j← 1 to C**

**7. if w[i]> j //背包放不下**

**8. dp[i] [j]= dp[i-1] [j]；**

**10. else**

**11. dp[i] [j]=max(dp[i-1][j],dp[i-1][j-w[i]]+v[i])；**

**12.return dp[n][C]**

**（3）算法时间性能：O（n\*C）**