**[Treap 查找第K小的数](http://www.byvoid.com/blog/treap-kth/" \o "Permanent Link to Treap 查找第K小的数)**

[計算機科學](http://www.byvoid.com/blog/cate/computing-science/" \o "查看 計算機科學 的全部文章) [Add comments](http://www.byvoid.com/blog/treap-kth/#respond)969 views

排序二叉树能解决好多问题，除了基本的插入、删除、遍历、查找键值以外，还有查找第K小（大）的数的操作。看了一天程序，总算明白了，其实本质和二分查找相似。

从根节点开始查找第r小的数，如果r<=(k的左子树节点的总个数)，那么第r小的节点一定在k的左子树，在k的左子树继续查找第r小的节点。  
如果r>(k的左子树节点的总个数+节点k的重复单元个数)，那么第r小的节点一定在k的右子树，在k的右子树继续查找第r-(k的左子树节点的总个数+节点k的重复单元个数)小的节点。  
如果以上都不满足，那么k就是第r小的数，返回k的键值。

（其实我很不习惯用静态数据存储二叉树，但考虑到效率和iRachex大牛的建议，还是使用了静态存储）

查找以k为根的树中，第r小的节点的值：  
  
int find(int k,int r)  
{  
if (r<=p[p[k].l].size)  
return find(p[k].l,r);  
else if (r> p[p[k].l].size+p[k].cnt)  
return find(p[k].r,r-(p[p[k].l].size+p[k].cnt));  
return p[k].key;  
}