[**BST:Skip list(跳跃表)**](http://www.fcicq.net/wp/?p=108)

August 9, 2006 9:50 · Filed under [SEO](http://www.fcicq.net/wp/?cat=18), [技术杂谈](http://www.fcicq.net/wp/?cat=16), [独特视角](http://www.fcicq.net/wp/?cat=12)

Skip List 上查找的基本思想是先从最高的 Level 层上查找，找到 key 所在的范围  
后，再从较低的层次继续重复查找操作，直到 Level 1。

严格意义上说，跳跃表不属于查找树，但它能以极其简单的实现提供二叉查找树的部分功能：插入和查找。  
跳跃表操作的时间复杂度期望为O(log n)，最坏情况下为O(n)，类似于快速排序与随机化Treap。

结构

跳跃表源于链表，例如：  
HEAD->2->4->6->8->10->12->NULL

对以上链表进行查找，最坏情况下需要考察n个节点。但是如过对每一个节点附加指针，比如增加2->6->10，那么查找操作只要考察n/2+1个节点。  
以此类推，如果我们能够使每个序数是2^i倍数的节点都有一个指针指向下一个序数是2^i的倍数的节点，那么理论上最多只要考察log n个节点。  
然而这样苛刻的要求不利于程序实现，于是我们放宽要求：将带有k个指针的节点称为k阶节点。将前面“每个序数是2^i倍数的节点都有一个指针指向下一个序数是2^i的倍数的节点”的条件改为每个k阶节点的第i个指针指向的节点至少具有i阶。这是一个容易保留的性质，而我们可以通过随机指定k来使一个节点阶数为k的概率为2^-k。

查找

在头节点从最高阶开始，沿着这个阶一直走，直到找到大于我们正在寻找的节点的下一个节点或NULL前停下并转到低一阶继续查找。当降至一阶停止时，或者已经找到，或者它不在这个表中。

插入

类似于查找，记录每一个使我们转到下一阶的节点。找到位置后将新节点拼接到表中——将每一个被记录的节点的相应阶的指针指向新节点，并将新节点的相应阶节点指向原来指针指向的节点。

删除  
Skip List 上的删除操作只需直接删除元素即可（包括指针的修整）。

ps:  
听说有许多人需要进行这一项:  
针对一个实例实现 Skip List 的动态演示(图形演示)

这确实是比较让人恶心的.  
做人家重复的工作真不是件好事情.