**[Treap](http://www.byvoid.com/blog/treap/" \o "Permanent Link to Treap)**

[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/#respond)700 views

NOI在即，我还不会Treap，这是一件多么恐怖的事情啊。今天下午在iRachex大牛的指导下，我学了Treap。不能算是会了，只是初窥门径而已。

开始看了许多介绍文章，还被误导了。因为有人用最大堆，有人用最小堆，我被迷惑了。还有就是左旋和右旋，有的人正好认识相反，我恰好看着两篇认识相反的文章在学习。

总结一下Treap

插入：  
按BST基本性质插入，生成**修正值**(有人叫**优先级、附加值、堆权值**)，并按照最大堆序维护修正码。  
向左子树插入返回后  
如果左子修正值大于根修正值，堆序被破坏，将根旋转到右子树（右旋）  
向右子树插入返回后  
如果右子修正值大于根修正值，堆序被破坏，将根旋转到左子树（左旋）

删除：  
叶节点：直接删除  
链节点：链接上子节点并删除  
完全节点：  
若其左子树修正值较小，将该节点左旋，递归删除左节点  
若其右子树修正值较小，将该节点左旋，递归删除右节点

左旋：将根X旋转到左子树

Y=X右子  
X右子=Y左子  
Y左子=X  
X=Y

右旋：将根X旋转到右子树

Y=X左子  
X左子=Y右子  
Y右子=X  
X=Y

一个非常棒的Treap演示  
[http://www.ibr.cs.tu-bs.de/courses/ss98/audii/applets/BST/Treap-Example.htm](http://www.ibr.cs.tu-bs.de/courses/ss98/audii/applets/BST/Treap-Example.html)

附一段第一次写的Treap代码  
#include   
#include   
#define MAX 100

using namespace std;

typedef struct  
{  
int l,r,key,fix;  
}node;

class treap  
{  
public:  
node p[MAX];  
int size,root;  
treap()  
{  
srand(time(0));  
size=-1;  
root=-1;  
}

void rot\_l(int &x)  
{  
int y=p[x].r;  
p[x].r=p[y].l;  
p[y].l=x;  
x=y;  
}

void rot\_r(int &x)  
{  
int y=p[x].l;  
p[x].l=p[y].r;  
p[y].r=x;  
x=y;  
}

void insert(int &k,int tkey)  
{  
if (k==-1)  
{  
k=++size;  
p[k].l=p[k].r=-1;  
p[k].key=tkey;  
p[k].fix=rand();  
}  
else  
if (tkey <P[K].KEY)  
{  
insert(p[k].l,tkey);  
if (p[ p[k].l ].fix>p[k].fix)  
rot\_r(k);  
}  
else  
{  
insert(p[k].r,tkey);  
if (p[ p[k].r ].fix>p[k].fix)  
rot\_l(k);  
}

}

void remove(int &k,int tkey)  
{  
if (k==-1) return;  
if (tkey <P[K].KEY)  
remove(p[k].l,tkey);  
else if (tkey>p[k].key)  
remove(p[k].r,tkey);  
else  
{  
if (p[k].l==-1 && p[k].r==-1)  
k=-1;  
else if (p[k].l==-1)  
k=p[k].r;  
else if (p[k].r==-1)  
k=p[k].l;  
else  
if (p[ p[k].l ].fix < p[ p[k].r ].fix)  
{  
rot\_l(k);  
remove(p[k].l,tkey);  
}  
else  
{  
rot\_r(k);  
remove(p[k].r,tkey);  
}  
}  
}

void print(int k)  
{  
if (p[k].l!=-1)  
print(p[k].l);  
cout << p[k].key << " : " << p[k].fix << endl;  
if (p[k].r!=-1)  
print(p[k].r);  
}  
};

treap T;

int main()  
{

int i;  
for (i=3;i>=1;i--)  
T.insert(T.root,i);  
T.print(T.root);  
for (i=3;i>=1;i--)  
{  
cout << endl;  
T.remove(T.root,i);  
T.print(T.root);  
}  
return 0;  
}