

```

1  /* Splay.h */
2  #ifndef _SPLAY_H
3  #define _SPLAY_H
4
5  //0--代表比较相同, >0--代表dataAddr<keyAddr, <0--代表dataAddr>keyAddr
6  typedef int SplayCmp(const void *keyAddr, const void *dataAddr);
7  typedef void SplayFree(void *);
8  typedef void SplayTraverseOp(void *);
9
10 typedef struct splay_node
11 {
12     struct splay_node *parent;
13     struct splay_node *lc;
14     struct splay_node *rc;
15     char key[0];
16 } SPLAYNODE;
17
18 typedef struct
19 {
20     SPLAYNODE *root;
21     SPLAYNODE *hot; //“命中”节点的父亲
22     int size;
23     int keySize;
24     SplayCmp *cmpFn;
25     SplayFree *freeFn;
26 } SPLAYTREE;
27
28 //Splay初始化
29 void SplayNew(SPLAYTREE *splay, int keySize, SplayCmp *cmpFn, SplayFree *freeFn);
30 //Splay销毁
31 void SplayDispose(SPLAYTREE *splay);
32 //Splay判空
33 int SplayEmpty(SPLAYTREE *splay);
34 //Splay规模
35 int SplaySize(SPLAYTREE *splay);
36 //Splay中序遍历（非递归）
37 void SplayTravIn(SPLAYTREE *splay, SplayTraverseOp *traverseOpFn);
38 //Splay中序遍历（递归）
39 void SplayTravInRec(SPLAYTREE *splay, SplayTraverseOp *traverseOpFn);
40 //Splay中查找关键码
41 SPLAYNODE *SplaySearch(SPLAYTREE *splay, const void *e);
42 //Splay判断某关键码是否在节点中, 返回值: 0--不在, !0--存在
43 int SplayFind(SPLAYTREE *splay, SPLAYNODE *node, const void *e);
44 //Splay中插入关键码
45 SPLAYNODE *SplayInsert(SPLAYTREE *splay, const void *e);
46 //Splay中删除关键码, 返回值: 0--成功, !0--失败
47 int SplayRemove(SPLAYTREE *splay, void *e);
48 //Splay中删除关键码（关键码非深度删除）, 返回值: 0--成功, !0--失败
49 int SplayRemoveU(SPLAYTREE *splay, void *e);
50 #endif

```