

Z题题解：块状链表（分块思想）

题意：维护一个数列，拥有插入和查询功能

本人很荣幸为这个题目贡献了一百次左右wa和tlim记录。

首先想到了用数组，查询只需要 $O(1)$ ；

但是插入却要 $O(n)$ ；

后来想到了链表，插入只要 $O(1)$ ；

但是查询却需要 $O(n)$ ；

怎么把这两个优缺点结合起来？

于是我想到一个办法，可以链表分成很多块，

形象点说，就是发派很多个排长，

他们每个人后面都跟着一定数量的兵；

当我要实现插入操作的时候，就可以一排一排的找，

先找到排，再找具体位置。

这样可以把单次查询的时间复杂度降为 $O(\sqrt{n})$

想法是很美好，但是我没有很好地实现它。

因为我是另开了一个数组记录排长的位置，

本质上我还是用连续的数组存的数据。

每次插入的时候，排长的位置都会发生改变，我还要全部更新一遍，

这就让我的程序跑的很慢。

后来接触到了块状链表这个数据结构，使我眼前一亮；

和我之前自己瞎想的方法相比，它有很多优秀的地方。

比如他是用结构体存的一个块，结构体成员size维护成员数目。这样就不必维护“排长”的位置了。

```
struct node{
    int next;//存下一个块的编号
    int size;//存当前块的大小
    ll mission[7000];//存任务
}no[maxn];
```

另外，块状链表还有删除，合并，分裂等功能，可以防止块状链表退化。

```

void split(int node){
    no[++cnt].next=no[node].next;//分裂，通过cnt分配新的节点
    no[node].next=cnt;
    int rest=no[node].size/2;//对半分
    no[cnt].size=no[node].size-rest;
    no[node].size=rest;
    if(no[cnt].next==0) tail=cnt;//更新尾指针
    for(int i=rest+1;i<=cut+1;++i){
        no[cnt].mission[i-rest]=no[node].mission[i];//说实话分裂这里挺耗时间的
        //不过从长远视角来看，及时分裂是好的选择
    }
}

```

本题只需要实现分裂功能即可，因为只插不删。

时间复杂度 $O(m*\sqrt{n})$:

单次查询和修改都是 \sqrt{n} , 一共要进行 m 次操作