**[树状数组](http://www.byvoid.com/blog/binary-index-tree/" \o "Permanent Link to 树状数组)**

树状数组是一个优美小巧的数据结构，在很多时候可以代替线段树。一句话概括就是，凡是树状数组可以解决的问题，线段树都可以解决，反过来线段树可以解决的问题，树状数组不一定能解决。

树状数组英文名称为**Binary Index Tree**，直译过来就是**二进制索引树**，我觉得二进制索引树更能说明其本质。树状数组的本质就是一种通过二进制位来维护一个序列前i和的数据结构。

对于维护的序列A，定义C[i]=A[j+1]+…+A[i]，其中j为i的二进制表示中把最右边的1换成0的值。j的值可以通过lowbit求出，即i-lowbit(i)。

lowbit(a)为2^(a的二进制表示末尾0的个数)。可以用下面式子求出

|  |
| --- |
| lowbit(a)=a&(~a+1) |

或者根据补码的性质简化为

|  |
| --- |
| lowbit(a)=a&(-a) |

修改方式如下

|  |
| --- |
| void modify(int p,int delta)  {  while (p<=N)  {  C[p]+=delta;  p+=lowbit(p);  }  } |

求前缀和如下

|  |
| --- |
| int sum(int p)  {  int rs=0;  while (p)  {  rs+=C[p];  p-=lowbit(p);  }  return rs;  } |

*[BYVoid](http://www.byvoid.com/) 原创讲解，转载请注明。*