**[树状数组](http://www.cppblog.com/qywyh/articles/19095.html)**

下图中的C数组就是树状数组,a数组是原数组;

可以发现这些规律:   
C1=a1   
C2=a1+a2   
C3=a3   
C4=a1+a2+a3+a4   
C5=a5   
……   
C8=a1+a2+a3+a4+a5+a6+a7+a8   
……   
C2^n=a1+a2+….+a2^n   
对于序列a，我们设一个数组C定义C[t] = a[t – 2^k + 1] + … + a[t]，k为t在二进制下末尾0的个数。   
K的计算可以这样:   
2^k=t and (t xor (t-1))   
以6为例   
               (6)10=(0110)2   
xor    6-1=(5)10=(0101)2   
                        (0011)2   
and          (6)10=(0110)2   
                        (0010)2   
  
所以问题变的很简单,重要写几个函数就可以了;   
求2^k的函数代码如下:

|  |
| --- |
| int Lowbit(int t)  {      return t & ( t ^ ( t - 1 ) );  } |

求1 -- end和的函数代码如下:

|  |
| --- |
| int Sum(int end)  {      int sum = 0;      while(end > 0)      {          sum += in[end];          end -= Lowbit(end);      }      return sum;  } |

对某位进行操作函数如下(以加法为例)

|  |
| --- |
| void plus(int pos , int num)  {      while(pos <= n)      {            in[pos] += num;            pos += Lowbit(pos);      }  } |

下面是用树状数组做的 pku star

http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <iostream>  
http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifconst int N = 32100;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint c[N];//树状数组  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint lowBit(int x) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return x & (x ^ (x - 1));  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid add(int x, int k) http://www.cppblog.com/Images/dot.gif{   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //a[x] += k  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (x <= N) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        c[x] += k;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        x += lowBit(x);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid sub(int x, int k) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //a[x] -= k  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (x <= N) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        c[x] -= k;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        x += lowBit(x);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint sum(int x) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //return sum(1..x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ret = 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (x > 0) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        ret += c[x];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        x -= lowBit(x);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return ret;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint out[N];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint f[N];  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint main() http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //pku2352  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i, j, k, x, y;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int n;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d", &n);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=0; i<n; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d", &x, &y);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        x++;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        add(x,1);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        out[sum(x-1) + f[x]]++;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[x]++;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=0; i<n; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        printf("%d\n", out[i]);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    system("pause");  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}