那么update函数就变成了下面的代码:

http://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifupdate  
 1http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid update(int x,int y,int val)  
 2http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
 3http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     int y1;  
 4http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     while(x < max\_x)  
 5http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif       http://www.cppblog.com/Images/dot.gif{//更改外面的一维数组  
 6http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif          y1 = y;  
 7http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            while(y1 <= max\_y)  
 8http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif               http://www.cppblog.com/Images/dot.gif{//对每个外面的一维数组所包含的一维数组进行更改  
 9http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                  tree[x][y1] += val;  
10http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                  y1 += (y1 & -y1);  
11http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif               }  
12http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            x += (x & -x);  
13http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
14http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif       }   
15http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}

同样的求和函数read()就会变成如下代码:

http://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifread  
 1http://www.cppblog.com/Images/OutliningIndicators/None.gifint read(int x,int y)  
 2http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
 3http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifint y1,ret;  
 4http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifwhile(x > 0)  
 5http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif   http://www.cppblog.com/Images/dot.gif{//求外面的一维数组的和  
 6http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     y1 = y;  
 7http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     while(y1 > 0)  
 8http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif       http://www.cppblog.com/Images/dot.gif{//对每个一维数组所包含的一维数组求和  
 9http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif         ret += tree[x][y1];  
10http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif         y1 -= (y1 & -y1);  
11http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif       }  
12http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     x -= (x & -x);  
13http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif   }  
14http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifreturn ret;//最后的结果  
15http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
16http://www.cppblog.com/Images/OutliningIndicators/None.gif