# 线段树

用线段树来维护各种高端操作

## Hdu6133

题意：给你一棵n个节点的二叉树，每个节点有一个提交任务的时间，每个节点总的提交任务的罚时为：提交这个节点和其子树所有的任务，每个任务提交时间的总和为该点的罚时。求每个节点提交完所有任务的最小罚时。

题解：根据题意，我们可以知道每个节点的提交的最小罚时为，按照任务的提交时间从小到大的来提交任务，可以得到最小的罚时。所以我们可以用线段树合并，先建立权值线段树，记录权值区间L到R的所有权值sum与size，线段树上的每一个非叶子结点的ans为ans[lchild]+ans[rchild]+size[rchild]\*sum[lchild]。

叶子结点i为size[i]\*val[i]

然后线段树u,v合并为u的时候我们把在保留u所有结点的同时，还要连上在v中存在而在u不在的点

Hdu6155

题意是给一个01字符串，然后有2种操作，   
1、把l到r这个区间的字符翻转，   
2、查询l到r这个区间有多少个不同的子序列，（注意是子序列，可不连续）

如果第i个字符为0，则dp[i][0] = dp[i-1][1] + dp[i-1][0] + 1，dp[i][1] = dp[i-1][1]

Dp[i-1][1] 加上这个1就会形成结尾为11的串，Dp[i-1][0] 加上这个1就会形成结尾为01的串，然后这个时候还要加上单个1这个串

如果第i个字符为1，则dp[i][1] = dp[i-1][1] + dp[i-1][0] + 1，dp[i][0] = dp[i-1][0]

于是构造一个3\*3的矩阵来维护这个转移关系就好了

然后用线段树维护每个区间的矩阵乘积就可以解决查询操作了。

对于修改操作，我们给区间维护一个flip标记，表示该区间是否要翻转，用线段树区间更新的方法去更新flip标记就好了。

将一个区间翻转后，它对应矩阵也要发生改变，这里我们只要将矩阵的第一列与第二列交换后再将第一行与第二行交换就好了。

注意这题目卡常数严重，因为最后一列一直是0 0 1所以就不要计算了