ZJOI2010 **基站选址 解题报告**

**Written by:Celery.T**

**【分析】**

动态规划。

设n表示村庄数量，m表示基站数量。d[i]，c[i]，w[i]，s[i]如题意所述。设st[i]=d[i]-s[i]表示可以覆盖第i个村庄的最小位置，设en[i]=d[i]+s[i]表示可以覆盖第i个村庄的最大位置。

设f[i,j]表示在第i个村庄建第j个基站，那么有状态转移方程：



为了方便处理，我们在最开始加入第0个基站，d[0]=-∞最后添加第n+1个村庄，d[n+1]=∞，我们约定一定在这两个村庄建基站，那么：，时间复杂度：O(n^3\*m)。

显然无法承受。

**【优化】**

我们设，则方程变成下面的形式： ，进一步的，因为在转移时i，j是不变的，所以我们把i，j省去，原方程就变成了这个样子：



这里出现了区间最小值的形式，也就是要取[j-1,i-1]中的最小的weight[k]，我们考虑用线段树维护weight[k]。

我们考虑决策k，设决策k对应的不被覆盖的村庄的集合为{x}，那么对于[j-1,k]这个区间内的决策，{x}内的村庄一定也不被覆盖。这个结论是显然的，因为d[]满足单调性。

我们设before[i]表示在第i个村庄之前不可以覆盖第i个村庄的最大位置，也就是说。那么显然的在村庄x之前不能覆盖村庄x的决策的范围就是[j-1,before[x]]，这时如果当前状态i也不能覆盖x，那么weight(j-1)..weight(before[x])我们都要加上w[x]。然后我们再取[j-1,i-1]中的最小值进行决策。before可以用二分查找求出。

既然x在决策区间[j-1,before[x]]不能被覆盖，那么 x在决策区间[j-1,before[x]]就永远不能被覆盖，只是因为，所以在weight(j-1)..weight(before[x])没有加上w[x]。但随着i的递增，d[i]也在递增，于是可能会出现一些不能被i覆盖的村庄，这时就又要在weight(j-1)..weight(before[x])加上w[x]。于是，我们把en[]按照递增顺序排序，在决策当前阶段时从1开始往后扫描，一旦发现en[x]<d[i]，就在weight(j-1)..weight(before[x])加上w[x]。这样每个x只会被算一次，均摊复杂度是O(1)。

这样优化后，时间复杂度为O(mnlogn)。