

## Q题题解：二分建边+Kruskal

题意：n个节点中任意两个节点*i,j*距离满足 $dis=A_i+A_j+D|i-j|$ ，其中A是各个点的初始常数，D是代价系数，求最小生成树。

最小生成树可以直接套用三种算法，Kruskal, Prim, Boruvka。-(吐槽：为子这道题专门学子Boruvka，但是没过；后来换的Kruskal，还是没过；最后发现是一个大于号写成小于号了.....)-

但是，考虑到这道题目的数据量之大 ( $10^5$ )，如果暴力建边的话恐怕内存空间不够 -(直接凡G子)-。

所以，我们可以想个办法，减少不必要的边的构建 (即那些构建生成树不可能用到的边)。

由题目给的权重计算公式：

$$dis[i][j]=|i-j| \times D+A_i+A_j$$

对这个公式就行稍加改造，我们可以得到:(这里我们默认j在i的左侧，故绝对值可以直接去掉)

$$dis[i][j]=(jD+A_j)+(-iD+A_i)$$

可以发现，公式中i和j是独立的 ( $j>i$ )。

这样的话，不妨把所有点分治处理，从中间分成左右两部分，只考虑左右两部分点之间的建边，左边点的点权为 $-iD+A_i$ ,右边的点的点权为 $D+A_j$ ;

左右任意两点之间的边权即为点权相加。

然后我们找到了左右点权最小的点，假设分别为min\_l,min\_r。

```
//假如说找左边最小点
for(int i=left;i<=middle;++i){
    point_weight[i]=popu[i]-i*cost_coe;
    if(point_weight[i]<min){
        min=point_weight[i];
        min_point=i;
    }
}
```

然后把min\_l与所有右半部分的点建边，把min\_r与所有左半部分的点建边。

```
//假如说从min_r建边
for(int i=left;i<=middle;++i){
    add(max_point,i,point_weight[max_point]+point_weight[i]);
}
```

这样循环递归下去，一共只需要建立 $n\log n$ 条边，在计算机的承受范围内。

```
//递归结构
void build_edge(int left,int right){
    if(left>=right) return;//递归结束条件
    int middle=(left+right)/2;//找中点

    build_edge(left,middle);//进入左边继续建边
    build_edge(middle+1,right);//进入右边继续建边
}
```

然后就可以用Kruskal了。

```
void kruskal(){
    for(int i=1;i<=n;++i) pre[i]=i;
    sort(edge+1,edge+tot,cmp);
    int cnt=0;
    for(int i=1;cnt<n-1;++i){
        int preu=find(edge[i].u);
        int prev=find(edge[i].v);
        if(preu!=prev){
            ans+=edge[i].weight;
            cnt++;
            pre[preu]=prev;
        }
    }
}
```

### 时间复杂度 $O(V\log V+E\log E)$ :

前一项是建边，要建 $V\log V$ 个边，后一项是Kruskal排序。