10.10NOIP模拟赛题解

A. 信号塔

所有的前缀 $\max$ 都可以成为信号塔，那么将每一段的分界线可能出现的不同位置的个数 $+1$ 乘起来就是答案（$+1$ 是因为这条分界线可以不存在）。

B. vivo50

题目大意：

给定 u, l, r，求 $\min \operatorname{dis}(u, v)$，其中 $v \in [l, r]$。

考虑到 $v \in [l, r] \Leftrightarrow v \in [l, m] \cup [m + 1, r]$，并且允许离线，因此可以先把所有的询问 ${l,r}$ 放到线段树上。

对于线段树上每个点区间，可以暴力建一颗虚树。

然后问题转化为了给定一个点 u，有一棵树，上面有一些点是关键点，求 u 到关键点的最近距离。

这就是个简单树上dp，算一算就好了（迫真）。

```
void go(int u, int fa) {
    if(u == root) h[u] = inf;
    if(spe[u]) h[u] = 0;

    the_top = 0;
    for(int i = head[u] ; i ; i = rest[i]) {
        int v = to[i], w = :: w[i];
        if(v == fa) continue;
        thev[++ the_top] = v;
        thew[the_top] = w;
    }

    int themn = inf;
    for(int i = 1 ; i <= the_top ; ++ i) {
        int v = thev[i], w = thew[i];
        h[v] = h[u] + w;
        h[v] = min(h[v], themn + w);
        themn = min(themn, f[v] + w);
    }
    themn = inf;
    for(int i = the_top ; i >= 1 ; -- i) {
        int v = thev[i], w = thew[i];
        h[v] = min(h[v], themn + w);
        themn = min(themn, f[v] + w);
    }
    for(int i = head[u] ; i ; i = rest[i]) {
        int v = to[i];
        if(v == fa) continue;
```

```
        go(v, u);
    }
}
```

总时间复杂度是 $O(n\log^2 n)$ 的。

当然也可以写树分块/莫队/边分治啥的……

by zhoutb:

写边分治可以只用二分和树状数组，利于卡常。点分治也行。

怎么有人开了 4s 把分块放过去了呢。

C. 选举

构造方法很多，其中一种是这样的：

```
999 501
2 1 2
1 2
1 2
2 2 1
2 3 4
2 3 2
2 4 5
2 4 3
2 5 6
2 5 4
2 6 7
2 6 5
2 7 8
2 7 6
2 8 9
2 8 7
...
```

D. 修路

算法一

暴力枚举每一个边的子集，判断是不是树，然后做树上DP求出答案。

复杂度 $O(C_m^{n-1}+n^{n-2}\times n)$ ，期望得分 30 分。

算法二

我们枚举每一个点，让他作为*中心城市*，并找一个*拥挤程度*最小的生成树。

我们发现 $\sum\limits_{i=1}^{n}dis(i,mid)$ 其实等于 $\sum\limits_{i=1}^{n}len_i\times sz_i$ ，其中 len_i 表示当 mid 为根时 i 到父亲的距离， sz_i 表示子树的大小。

设 $f[i][j][s]$ 表示 i 的子树，有 j 个叶子， i 的子树中点的集合为 s 的最小贡献。贡献是指 $\sum len\times sz$ 加上 $j\times S$ 。

转移时枚举一个新的节点 k ，让它做为 i 的儿子，然后再枚举一个 s 的子集 t ，让 t 成为 k 的子树，讨论一下即可。

同时要注意好根节点 mid 可能也是叶子。

复杂度 $O(3^n \times n^4)$ ，期望得分 50 分。

算法三

发现存在一些数据，保证 $m = \frac{n\times(n-1)}{2}, S = 0, l_i = 0$ 。

这表明任意一个规划的*拥挤程度*都为 0 。

那么方案数即为生成树个数乘以 n ，即 $n^{n-2} \times n = n^{n-1}$ 。

算法四

发现存在一些数据，保证 $m = \frac{n\times(n-1)}{2}, S = 0, l_i = 1$ 。

这表明我们修路必须是菊花图，然后*中心城市*要是菊花的中心。

那么最小的*拥挤程度*即为 $n-1$ ，方案数即为 n 。

算法五

发现存在一些数据，保证 $m = \frac{n\times(n-1)}{2}, S = 10^9, l_i = 1$ 。

这表明我们修路必须是一条链，然后*中心城市*要是链的中间。当然当 n 为偶数的时候中间的两个点都可以。

那么最小的*拥挤程度*即为 $2 \times S + \text{算} - \text{算}$ ，方案数即为 $\frac{n!}{2} \times (n\&1?1:2)$ 。

算法六

尝试优化算法二。

我们发现叶子的个数其实等于 $1 + \sum\limits_{i=1}^{n} \max(son_i - 1, 0) + (\text{根节点是叶子})$ ，其中 son_i 表示 i 的儿子的个数。

这个式子也就是“多岔出来的边的个数” $+1+(\text{根节点是叶子})$ 。

所以我们省掉“叶子个数”这一维，然后在枚举 t 的时候，如果 $s \oplus t \neq 2^i$ ，那么直接给代价加上 S 即可，然后还要特判根节点。

$s \oplus t \neq 2^i$ 的意思就是这一次给 k 分配的节点没分配完，还剩一些（这些就不在 k 的子树里了），所以相当于“多岔出去一条边”，就是多了一个叶子。

复杂度 $O(3^n \times n^2)$ ，常数较小。期望得分 100 分。