gym103708B

记录f(u)表示u距离最近的antenna的距离。每一次加一个新antenna，从新antenna开始更新最短路数组f(u)。

如果只允许f(u)的取值是0, 1, 2, …, k以及inf表示太远了管不到，那么每个f(u)只会有k+2个可能取值，也就是总的更新次数是n\*k。

也就是BFS+剪枝的复杂度就是对的。



gym101615D

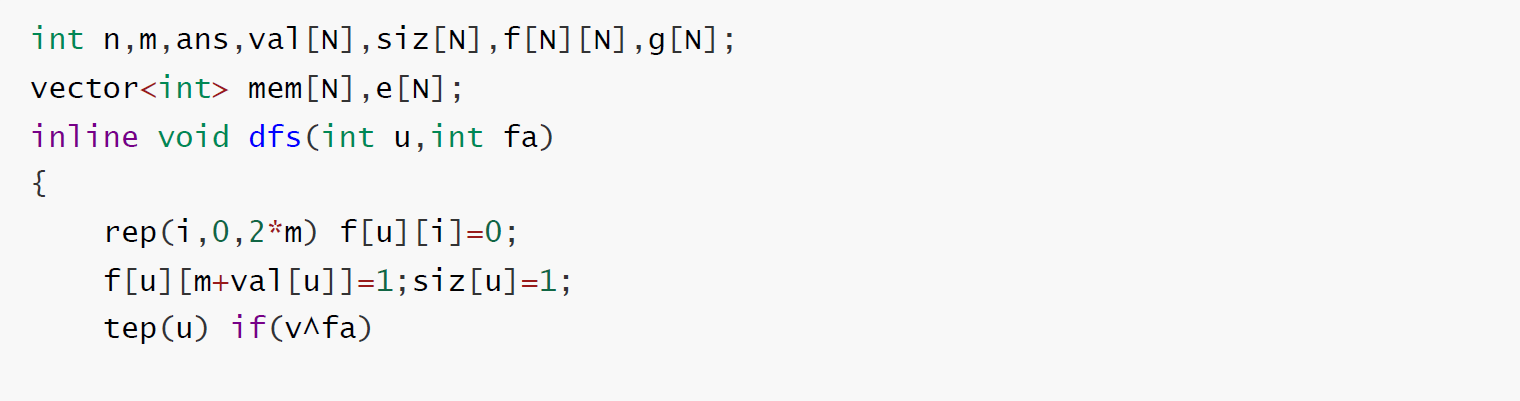
所有祸根来自于一个点u，它满足u的相邻边有两条是同色的，这回使得这两条边向外放射出去的部分（树上的一个连通块）不是好节点。

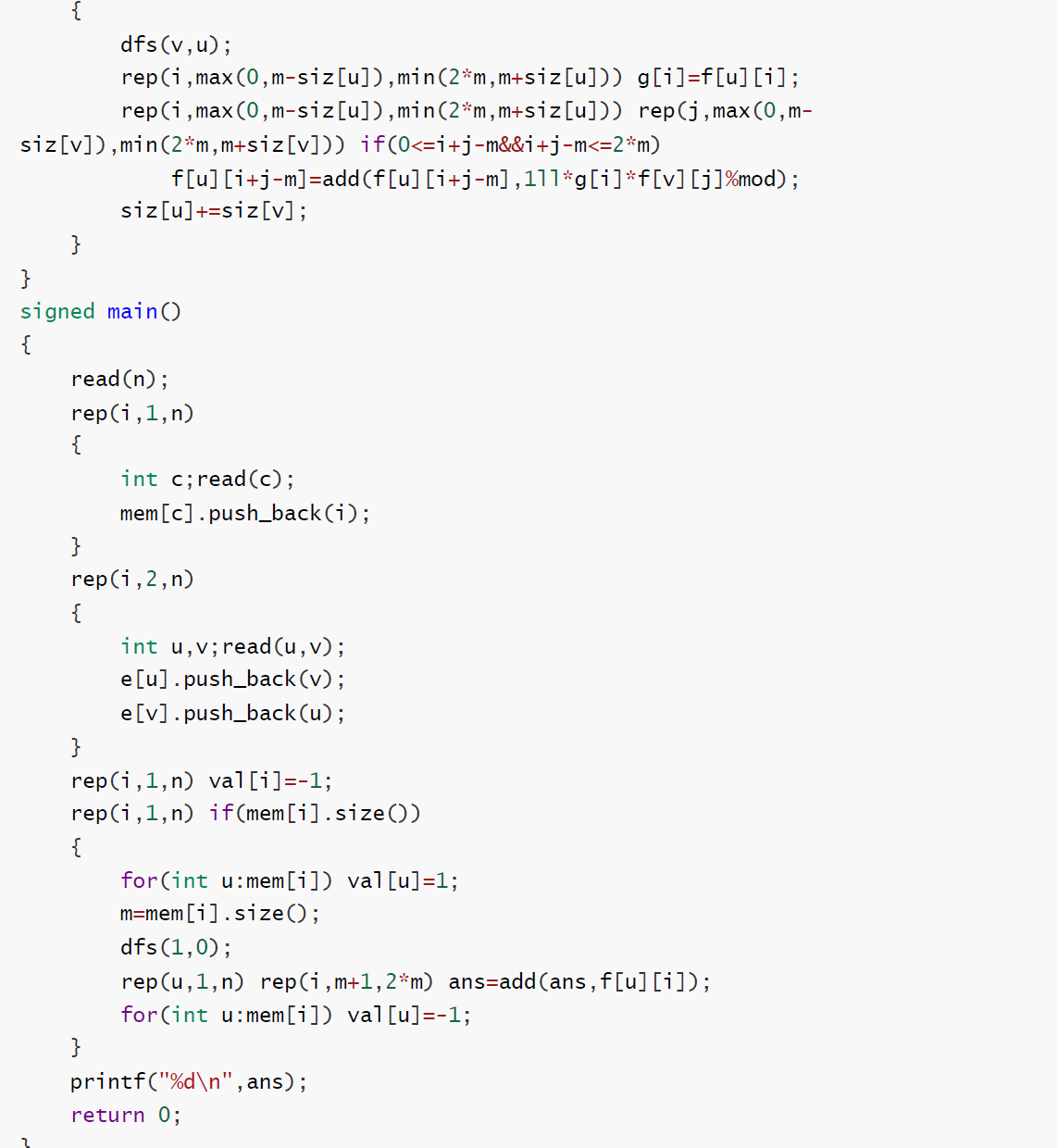
那么枚举所有u，然后对u造成的坏节点打标记（坏值+1），那么就是要支持子树+1，以及最终询问每个节点的坏值是否是0。

两种做法：

1. 找一个祸根点当根，每次ban掉的都是一个子树，可以证明最后的好点集合是一个包含根的连通块。直接DFS就行。特殊情况是某个子树里面存在一个祸根节点x想要标记x父亲方向的连通块都是坏的，那么整棵树都是坏的。
2. 树上差分。子树+1就记录在子树根上，父亲方向的连通块+1就拆成全体+1以及以我为根的子树-1。然后再做一次dfs把这个+1, -1的标记传递到每个节点。

都是O(n)。





gym103438I

首先注意到重要的概念是，黑的/白的连通块，而不是n\*m的网格。

对于每个连通块建点（下称基本点），连通块之间的相邻关系建立边，这就得到一个二分图。

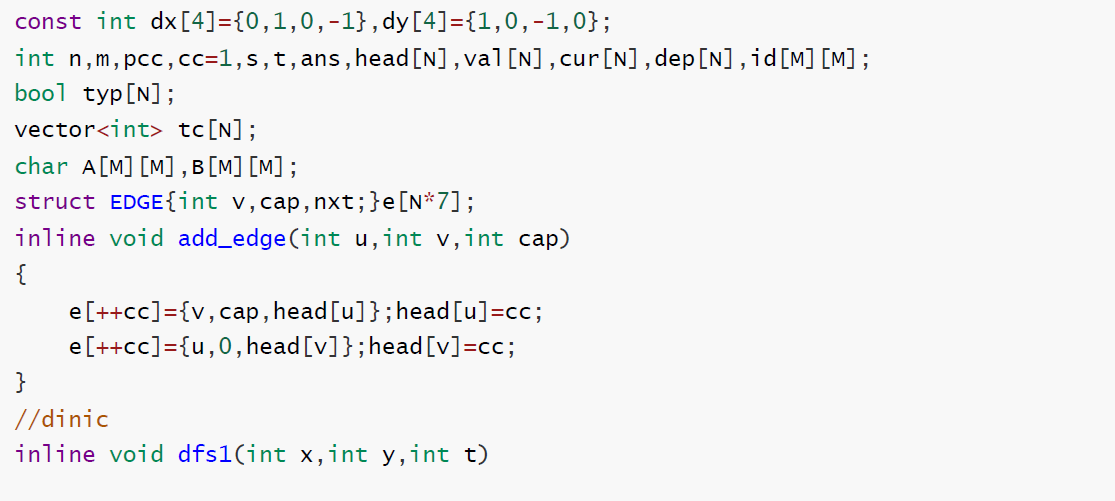
首先：一定有，每个基本点最多被反转一次颜色。证明略。

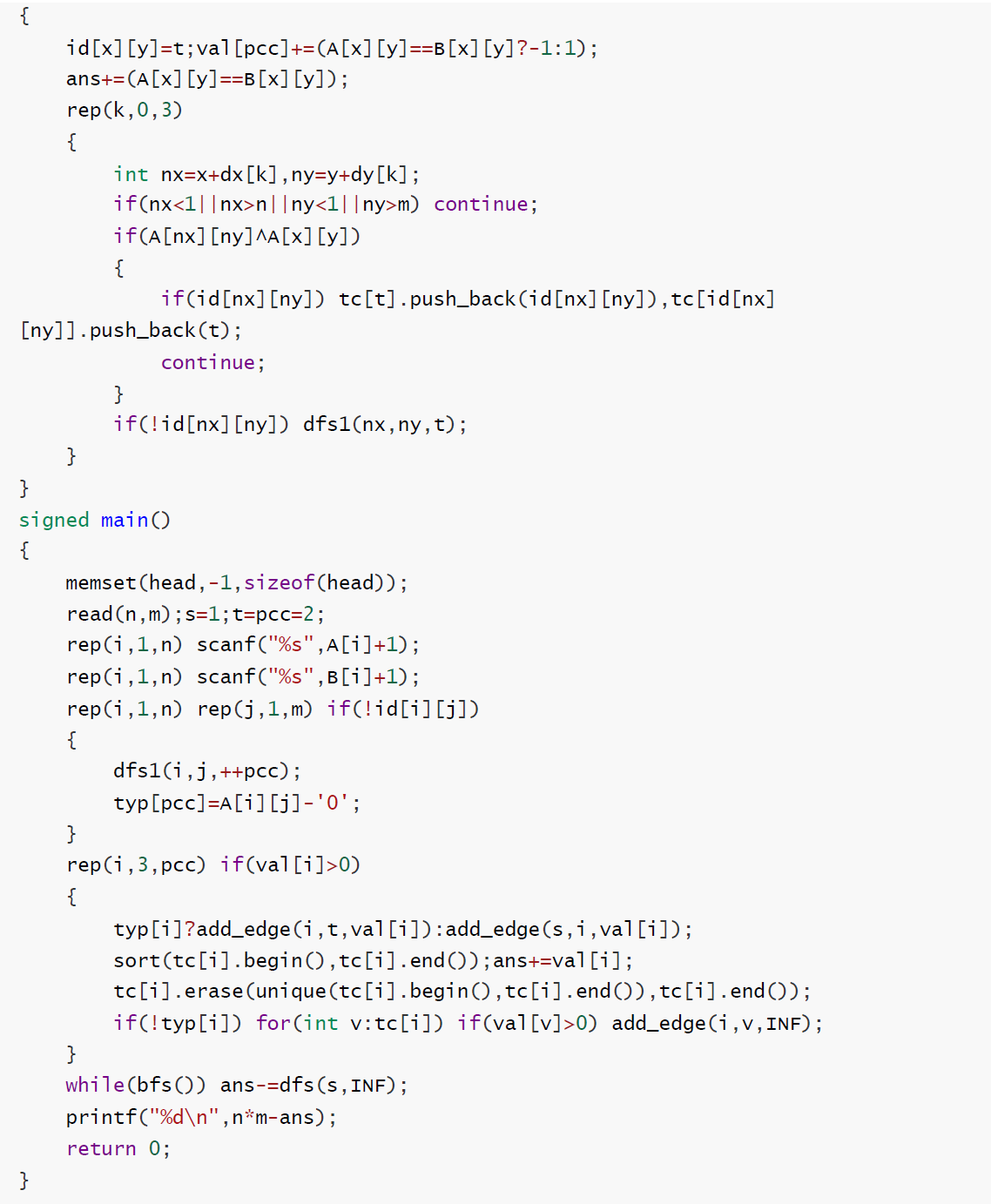
所以：操作序列一定可以形如，对于基本点图，不会翻转那些合并起来的大连通块。例如01 -> 11 这步ok，但是如果接着做 11->00这就不行因为你操作了合并产生的新连通块”11”。01->00的操作序列应该被标准化为flip第二个位置的1这一个操作。

所以操作就是在基本点图上选一个独立集（选出一些基本点，翻转它们；选的点们不相邻）。

翻转一个基本点的代价容易计算。【它保持原样对答案的贡献（跟B的差） 和 翻转之后的贡献 这两者之差】

化简成为二分图最大权独立集。模板题。





gym100962F

树上莫队模板题。去oi-wiki可以学。

首先自己去学一下区间莫队。

树上的路径可以是区间，如果取树的括号序列。那么u->lca(u, v)->v的这条路径上，u->（lca的u侧的下面一个点）对应右括号，（lca的v侧的下面一个点）对应左括号，lca不出现，中间跳过的那些点一定完整出现了左括号和右括号。

那么每次莫队插入/删除一个括号的时候需要判断这是区间内唯一的括号，还是区间内存在一个和它呼应的同一个节点的另一个括号，来判断到底要加上这个括号的节点的信息还是减去。

记得询问的时候人为的加上一个lca节点的信息。

这样莫队n^1.5次修改+n次查询，每次都set操作维护mex可以做到n^1.5\*log。

优化是用分块数据结构，修改O(1)查询O(n^0.5)，非常匹配莫队。类似线段树不过每个点有n^0.5个孩子，只有3层。具体看下面的代码。

