# Travel题解

**题目大意**

现在有N个人，每个人选择A有Ai种方案，选择B有Bi种方案，问至少有C个人选择A的方案数为多少

有P组修改，每次修改一个人选择A与选择B的方案数

**题解**

主要思路：线段树维护dp。

考虑到C比较小，我们补集转化一下

若没有修改，考虑直接DP

用dp[i][j]表示在前i个人之中，有j个人选择了A的方案数

转移：dp[i][j]=dp[i-1][j-1]\*Ai+dp[i-1][j]\*Bi

此时考虑到，人的排列与答案没有关系，即我们可以先计算后面，再计算前面

考虑F[j]表示前一半人中，有j个人选择了A的方案数

G[j]表示后一半人中，有j个人选择了A的方案数

转移：dp[i]=F[j]+G[i-j] (0<=j<=i)

考虑修改，我们将第一个dp看做一个C\*N的数组，每C个为一列，一共N列

对N列使用线段树进行维护即可，发现总方案数亦可以维护，本题解决

# Map题解

**题目大意**

给定一颗仙人掌，请回答以某个点为根的子仙人掌中，权值的出现次数为偶数和为奇数的个数。

**题解**

主要思路：序+莫队算法。

考虑仙人掌的一颗序生成树。

不难发现一颗子仙人掌所对应的连续的一段或两段。

每当我们在一个环上时，如果我们优先走环边的话，我们会发现：之前的两段序变成了一段序+一个点。

这一段序有可能包含那个点，也有可能不存在。

事实上，如果我们优先走非环边，序就成为了连续一段了。

于是问题转化为序列上的问题。即每次询问一段区间内权值且符合出现次数条件的权值个数。

经典的莫队算法，用树状数组维护答案的前缀和即可。

由于是从仙人掌上演变而来，莫队算法很难达到复杂度上界，因此不需要对块大小做调整。

# Connection题解

**题目大意**

求一张图的边连通度。

**题解**

主要思路：网络流+分治。

确定将某两个点断开的算法为最大流算法。

则枚举断开哪两个点，可获得70分。

每次随机两个点进行计算，并找出方案，则两个联通块之间的最小代价已知。

只需对两个联通块之间的点进行分治枚举即可。