**A. Another LIS**

本问题可以分成两部分做：先为每个数找它在序列中的最终位置，然后再求每个数以自己结尾的LIS。

注意到一个事实，最后一个元素所插入的位置Xn正是它在最后序列中的位置，而正因为它的插入，在之前的1~N-1里位于Xn以及Xn之后的数会整体向后移一位。如果我们将整个插入的过程倒过来变成删除的过程，初始N个位置全被占据，将元素N删除之后Xn变成了空位，那么对于N-1来说，它的位置应该是那个N-1个依然被占据的位置中的第Xn-1个，接着将N-1删除，同样可以得到N-2的位置。如此递推即可。过程中有查询第k个数的操作可以使用线段树或类似结构实现。

第二部分的问题其实就真的只是裸的LIS了，方法很多在此便不赘述。

**B． Bi-peak Number**

按位DP的题目，需要上下界同时考虑。dp[i][D][U][last][state]前i位，D记录下界是否达到，U记录上界是否达到，last记录最后的数字，state记录当前状态。

其中state可以用一个7个节点的状态机来表示。分别对应如下状态，注意状态间的转移即可。

6

0

1

2

3

4

5

**C． Counting Heads**

看来大家都很honest，开场那么久都没人过。。。

此题灵感来自于百度之星决赛晋级规则。

对于两次成绩，枚举第一场进总决赛的最低排名为i，第二场最低排名为j，枚举完后，可以确定两场排名都在晋级范围内的人数为a，只有第一场在晋级范围内的人数为b，只有第二场在晋级范围内的人数为c，则可以确定在第一场进总决赛的最低排名为i，第二场最低排名为j的条件下，这a个人是一定能进入决赛的（因为不管取哪场成绩都在范围内）。这时我们只需枚举在a个人中有多少个人以第一场成绩晋级，有多少个以第二场成绩晋级，假设有k个人以第一场成绩晋级，则第一场还可以有g-k个晋级名额，则将这g-k个晋级名额尽可能多给本国选手，第二场还可以有g-(a-k)个晋级名额，再将这g-(a-k)个晋级名额尽可能多的给本国选手。这样复杂度是N^3，

此题还有N^2，N^2logN的做法，详见标程。

**D．Eight II**

经典题目变型。难度不大。各种方法皆可：枚举空位（3种或9种）暴力打表，置换后反向求解路径；双向BFS，正向照常，反向DP，小心处理接口处；A\*或IDA\*亦可。

E．Help Johnny

作为水题，它很成功。O(n)预处理一下即可，不过没想通为什么那么多队敢于交N2的暴力­­­。。。- -！

F． Imaginary Date

同为水题。。。k <= n，所以每一步都可以取得巧克力。而每一步取到某个巧克力的概率是相等的。所以此题答案与k无关。故答案是sum(v)/n\*m。

G． More Happiness

题面描述改过很多遍了，不过大家读起来可能还是有点囧，比赛的时候也有很多人问，可是我真的是把条件都说清楚了的啊T T。。。

题意是给定了花费限制，树的边上会有花费，点上也会有花费（不过只算一次），要求在树中找一条路径经过尽可能多的点。比赛开始后我们才发现描述中的whenever那句话完全可以不要的，因为最优策略是不会跳过某个经过的点不选的。由于查询很多，所以必须要预处理。再者，因为点上的花费是可变的，因此在预处理的时候不能将点上的花费作为状态。

典型的做法是树形DP ，需要知道以x为根的子树T(x)中的以下三种情况的值：

1. V[m][0]： 路径起点和终点都在T(x)外且经过x的在T(x)中经过m个点路径的最小花销;
2. V[m][1]： 路径起点和终点不同时在T(x)外且经过x的在T(x)中经过m个点路径的最小花销；
3. V[m][2]： 路径起点和终点同时在T(x)外且经过x的在T(x)中经过m个点路径的最小花销。

可知，求出所有点的V[m][2]后对于给定m求最小值，那么对于点上花销为X的查询，只需去找寻满足条件的最大的m即可。

上述树形DP是不需要枚举起点的，转移过程中需要用到背包，总的时间复杂度为O(N^3)。如果选择枚举起点的做法的话，就必须要使用O(N^2)的树形背包解法了，具体可以参见国家集训队论文。

另外，利用分治和上述的定点O(N^2)树形背包可以将复杂度降为O(N^2logN)。

H．N-dimensional Sphere

类比于二维，三维的情况，此题容易想到解一个线性方程组。每个方程表示到两个N维点的距离相等的N-1维平面（点法式很好写），选择N个线性无关的方程联立解之即可。

不过考虑到精度要求很高(17位），所以用double是恒定WA的。因为所有坐标及答案都为整数（球心满足|Xi|<=10^17,这个最开始出题的时候没注意到必须限制，比赛中有人问到才发现，实在失误了）可以取模求解。找一个比2\*10^17大的素数取模，可以实现 +,-,\*,/ 运算，写个高斯消元，最后把解还原到|Xi|<=10^17即可。

高斯消元复杂度为O(N^3)，但是因为有O(N^3)次long long \* long long 取模的运算，比较标准的写法（像快速求幂那样）为O(lg V)(V为数值大小），所以最后的复杂度应该为O(N^3\*lgV) (这道题V为你设的模数)。

I．Task Schedule

这个题原本时间规模是5000，后来改成了500，结果内部验题的时候即被plt用一个500^2\*log500的贪心搞了过去-\_-。我的做法是网络流。左边N个点，每个点表示一个任务，源点向这N个点分别连一条边，流量为Pi。将每个任务的起始时间和结束时间一起排序，可以得到一个时间序列，从而得到M个时间区间。如果左边第i个点的处理区间和右边第j个点代表的区间有交集，则连边，流量为交集大小。右边M个点每个与汇点连边，流量为M。如果这个图的最大流可以使源点到左边N个点的每条边满流，则存在可行方案，反之不存在。