Problem provider: [**isea**](http://bestcoder.hdu.edu.cn/rating.php?user=isea)

**1001 Revenge of Segment Tree**

求一个序列的所有连续子序列的序列和的和。

考虑每个数出现在多少个子序列之中，假设第i个数为Ai，区间为[L,R]。那么包含Ai的区间满足L⩽i⋂R⩾i。累加(L+1)∗(N−R)∗A[i]就可以了。

当然不能上来用个线段树无聊大家，是吧。PS. I'm a ds-antier QAQ

复杂度：O(N)

难度：0.5/5

**1002 Revenge of LIS II**

求序列第二长的上升子序列。

看上去输出第二大的LIS就可以了。嗯，1 1 2可以hack不少了。

LIS的O(N2)做法是dp[i]=max(dp[j]|a[j]<a[i])+1。如果需要第二大的，再额外DP一个至少有两个不同递增序列结束在i的最大长度。那么dpdirty[i]=max(dpdirty[j]|a[j]<a[i])+1，或者dpdirty[i]=max(dp[j]|Count[dp[j]]>1)+1。

这个思路不是很容易推广到O(NlogN)的LIS算法中，诸位可以思考一下。

复杂度：O(N2)

难度：1.5/5

**1003 Revenge of Nim II**

Nim游戏的后手作弊移走一些整堆的物体（不能全拿走），可以保证先手必败吗？

Nim游戏先手必败的条件是XORSum(a[i])=0。后手的目的就是找到这样的一个非空子集。把这里的a[i]看做一个每位为0或1的行，所有的数字组成一个矩阵，矩阵空间的运算是XOR。如果这个矩阵满足性质Rankmat=RowNummat，那么它的任意一个子集的XORSum都不相等，且非空子集的XORSum不为0，否则矩阵的Rank就会小于RowNum。

如果Rankmat<RowNummat呢？那么对于某个子集满足RankSubmat<RowNumSubmat，其他的行可以由这个矩阵的某些行XOR得到，就一定存在一个XORSum(a[i]|a[i]⊂subA)=0的subA存在。

复杂度： O(MAXB3)|MAXB=log(MAXA)≈40

难度：2.0/5

**1004 Revenge of iSea**

给出N道难度递增的题目，难度用可能做出的百分比表示，选出K道题目使得做出K-1道题目的概率最大。

假设答案是{PC[i]}，做出k-1的概率为∑1≤i≤k(1−P[PCi])∗∏1≤j≤k⋂j≠iP[PCj]。直接尝试转化这个式子的效果并不明显。

换个思路，假设最优解已经包含了k-1个了，现在来选取最后一个。K-1个全部做出的概率是Pall(k−1)，有一道为做出的概率是Pless(k−1)，现在选取的是PCk，那么做出K-1道的概率是

Pall(k−1)∗(1−P[PCk])+Pless(k−1)∗P[PCk]=

Pall(k−1)+P[PCk]∗(Pless(k−1)−Pall(k−1))

这是一个关于PCk的一次函数，如果Pless(k−1)−Pall(k−1)为正，选取最大的PCk，否则选取最小的。

这样，可以证明答案一定选取两边的概率，枚举比较一下就可以算出最大的概率了。

还有最后一个问题，需要求字典序最小的。对于左边选取的Pi，当然index越小越好，对于右边的，如果存在相同的value，应该选取index较小的。比如90 80 30 30，如果答案是第一个和最后一个，为了取得最小的字典序，需要用第三个来替换一下第四个。

复杂度：O(K3)

难度：2.5/5