第一道题只要发现一个规律：把所有CAT以外的元素删掉，如果不是CAT就是IMPOSSIBLE。

第二道题没来得及提交，我的做法是回溯，2^50复杂度，做题速度太慢。

BBS上问人后得到的答案，

将原题看作：每只猫向右跳为跳2X，向左跳为静止。

最优解具有以下形式：前k个猫向右跳，后n-k只猫向左跳。

就是能证明一个结论：设一个解里面k是最右边一个向右跳的猫，那么前面的任何一只猫向右跳都只会使左边界减少而不会使右边界增加。

第三道题压根没来得及看。

补充：

1000pt的题目比较难。

首先要看出，这是一个背包问题的变种（简直是变异）。

也就是，对应某种状态，求解选择1…i个这中状态的情况个数，选择(ai,bi)后，余下的状态可能的情况个数+不选择（ai,bi），余下的状态.

C[i][S]=C[i-1][S]+{选择(ai,bi)后的S的每种可能S’的C[i-1][S’]之和}

如果S是这些所有的数所有可能组成的集合，那么，S就是一个50bit的数，如此的状态可能是2^50，那么DP需要2^50\*50，显然会超时，而且可能也没有这么大的数组。

于是需要压缩S的状态，这里技巧性很强。

因为1…10是可以重复的，因此选了(ai,bi)在子问题里面还要考虑是否这些数可能被选中，因此这些数的状态必须记下来s。

而11…50这些数是不会重复的，所以可以只记录这些数字的个数r，按照个数将情况区分。

于是，一个状态变成(s,r)，s是1…10可能的配置，r是有多少个{11…50}里面的数。

这样c[i][(s,r)]=c[i-1][s,r]+{选择(ai,bi)后，1…i-1可能组成的每种状态的情况数}

在i时，选择ai,bi后，假设子状态为(s’,r’)，则r’=r-1。而s’的可能性最多有四种，s,s-{ai},s-{bi},s-{ai,bi}，这四种去重复后，就是选择(ai,bi)后的子状态，累加即可。