算法一

随便暴力。

根据不同的实现方式可得0~25分。

算法二

我们称呼两个人分别为A和B。

可以发现不在环中的边最后一定会被选，只有环中的边都选完了才会选树上的边（否则相当于把先手权利丢给对方，不优）。

所以只需要考虑每个环。

对于一个环，最后一定会剩下一条边，并且由于选不超过环长个数的边的时候一定不会形成环，所以相当于：给一个集合S，每次从S中取一个数，直到$|S|=1$为止。

对于某个S，显然A一定会选最小的，B一定会选最大的。将S从小到大排序，A从左边开始选，B从右边开始选。

对于每个集合S_i，记下左边和右边分别有多少个已经被选了，直接状压一下。DP的时候枚举当前这个人操作哪个集合。

对于Subtask3，有效的状态数为$O(n^3)$，于是可以做到$O(n^3)$的时间。$O(n^3\lg n)$也可以。

期望得分25分。

算法三

对于所有环长度为奇数的情况，有结论：S_i中最终没有选的是S_i的中位数。

证明：归纳。假设A选了S_i，B选了$S_j, j\neq i$，那么A就可以跟着B选（即下一步选S_j，然后B选什么A选什么），直到某一次B选到了S_i，接下来的是个子问题。所以如果B选了S_j，A有办法构造出和B选S_i一样的结果，即剩下的都是中位数，当然也有可能构造出对于B来说更劣的结果；而如果B一直跟着A选，最终也可以构造出剩下的都是中位数的情况。所以B为了没有更劣的可能，他一定会跟着A选。

于是每次选都是B跟着A选，最终剩下的当然是S_i的中位数。

通过这个结论可以过Subtask4。

算法四

对于所有环长度都是2的情况：对于一个二元组，A肯定选小的，B肯定选大的；那么每次选的人希望选择一个二元组，使得自己操作的收益和对方操作的收益相差最大。

贪心做，直接作差，排序，两者争着选差值最大的。

通过这个结论可以过Subtask5。

算法五

从算法三和算法四的结论推广而来。

如果所有的集合中只存在一个大小偶数的集合（记为S_1），其它的集合都是奇数：假如A操作S_1，后面变成了算法三的子问题，先后手顺序交换；假如A不操作S_1，那么B可以一直跟着A选，直到某一次A选了S_1。于是B有能力构造出和上面一样的情况，因此B也有可能构造出比上面还劣的情况（对A来说）。为了制止这种可能性的发生，A一定操作S_1。

如果存在多个大小为2的集合：用和上面类似的分析思路，得出A一定操作大小为2的集合。根据算法四的结论，A一定选择差值最大的。

继续推广到更复杂的情况，可以得到结论：A和B一定先操作大小为偶数的集合，并且争抢集合中位数（中间两个数）差值最大的。

归纳，如果A选大小为偶数的集合，根据子问题的结论可得A相当于选了较小的中位数，所以A一定会选择中位数差值最大的集合；如果A选了大小为奇数的集合，类似于上面的分析，B也可以造出和上面一样的情况，所以有可能构造出更劣的情况（可能存在的疑问：是否有能力一直跟着，即不存在A选了个大小为2的集合，且这个集合不是中位数差值最大的集合；然而由于A一定选中位数差值最大的集合，所以这种情况不存在），因此A一定先选大小为偶数的集合。

通过这个结论可以得到100分。