人赢

这是一道好题。

算法一

我会爆搜！

（不过爆搜好像挺难写的，反正我是没写过

预计得分5分

算法二

我会状压！

用 $dp_{S,i,j}$ 表示当前已经考虑了集合 S 中的所有元素，当前未完成的环的起点为点 i ，终点为点 j 时的答案。$dp_{S,0,0}$ 表示当前考虑了集合 S 中的所有元素，并且恰好拼成了若干完整的环时的答案。

复杂度 $O(2^n \times n^3)$，预计得分15分

算法三

我是图论带师！

所有点恰好在环上意味着每个点的度数都为 2 （自环也看做度为2），那么如果给边钦定方向，则可以让每个点的入度为1，出度为1.所以我们可以给每个点钦定一条出边，并且所有出边的终点互不相同。也就是说对于一个 n 个数的排列 p ，我们可以令点 i 的出边为 p_i ，从而形成若干个环。

而排列 p 又可以与一个二分图匹配一一对应，所以可以二分答案，然后通过检查是否存在二分图完美匹配的方式来得到最终答案。

复杂度 $O(n^3logv)$（匈牙利）或 $O(n^2\sqrt{n}logv)$（Dinic），预计得分40分

算法四

对于 $a_i = i-1$ 的包（n 没有限制），这可能是唯一一个跟正解搭的上边的包。

假设所有 a_i 的最高位位数的最大值是 k 。

我们可以根据第 k 位是 0 还是 1 来将所有点分成两部分，很显然有第一部分的点数大于等于第二部分的点数。如果等于，则说明 n 是 2 的幂，那么答案显然是 $n-1$ ，所以我们考虑不等于的情况。

假设第一部分的点数为 a_1，第二部分的点数为 a_2 （$a_1 > a_2$）。

由于答案的第 k 位一定是 0，所以一二两部分之间的边（也就是边权第 k 位为 1 的边）的具体权值对答案没有影响。（接下来**用中间边指代两部分之间的边**）事实上，我们可以贪心的让中间边尽可能多。因为对于任意一种中间边数小于 $2a_2$ 的情况，由于每个部分的度数和必定是偶数，所以中间边数必定也是偶数。那么**我们就能断开一条第一部分的边，断开一条第二部分的边，然后连上两条中间边，这样答案的变化一定是不劣的。**

每个点度数为2是满足题目要求（恰好形成若干不交环）的充分必要条件

在拉满中间边之后，第二部分的度数和已经等于点数的两倍了，而第一部分的度数和还小于点数的两倍。**我们将某一部分点数的两倍减去其现有度数称为这部分的剩余度数。**第一部分的剩余度数为 $2(a_1-a_2)$ ，也就是说第一部分内部还需连 a_1-a_2 条边。而在第一部分内，我们能够连出 a_1 条边权为 a_1-1 的边，同时保证每个点的度数不超过 2 ，即在点 i 与 点a_1+1-i 之间连两条边。所以第一部分内部连边的答案为 a_1-1 ，也就是整体的答案为 a_1-1。

复杂度 $O(n)$（用于判断子任务），预计得分 10 分。

算法五（正解）

一些名词解释与引理:

答案：指被最大化的边权最小值

连 k 条合法边：指连 k 条边后**每个点度数均不超过2**

部分：表示一个点集

最高位：指二进制下最高位，我们认为最低位为第 0 位

组：两个部分构成的一个集合体（**pair<部分，部分>**）

中间边：一个组中两个不同部分之间的边

内部边：端点在同一个部分内部的边

剩余度数：对于某个已经被连了一些边的部分，这部分的剩余度数表示点数的两倍减去其现有度数

引理1：每个点度数为2是满足题目要求（恰好形成若干不交环）的充分必要条件

引理2：只要中间边的边权的最小值大于两个部分内部边的边权的最大值，则选取尽可能多的中间边一定是不劣的（证明可以参考算法四）

我们考虑贪心地最大化答案

我们按位考虑，如果所有数的这一位都是 0 或都是 1 ，则继续考虑下一位。

也就是说我们现在找到了最高的一位使得所有数的这一位既有 0 又有 1

我们将全部的 n 个数分为两个部分，大小分别为 a 和 b 。根据算法四中的证明，我们需要最大化中间边的数量。如果 a 不等于 b ，则大小较小的那个部分的度数会被拉满（剩余度数为0），这意味着我们不需要考虑那个大小较小的部分。对于大小较大的部分，我们需要连一些内部边，使得该部分的剩余度数为0。这变成了一个与原问题类似的子问题，即在某个点集中连 k 条合法边，最大化边权最小值。我们称这个问题为**一类子问题**，用记号$f(S,k,v)$表示当前考虑到二进制下第 v 位，在点集 S 中连 k 条边的答案。

还有 a 等于 b 的情况，也就是说答案的这一位为 1 ，每条边都必须是中间边。我们称这个问题为**二类子问题**，我们待会儿再讨论它的做法。

一类子问题 $f(S,k,v)$:

根据第 v 位的值为 0 或者 1 ，将 S 分为两个部分 A 和 B ，其大小分别为 a 和 b 。如果 $k>2min(a,b)$，则说明答案的这一位为 0，需要先最大化中间边的边数，然后递归计算 $f(max(A,B),k-2min(a,b),v-1)$。如果 $k\le 2min(a,b)$，则说明答案的这一位为 1，连的边只能是中间边，那么我们称此为二类子问题，用记号 $g(T,k,v)$ 表示当前考虑到二进制下第 v 位，在组 T 中连 k 条中间边的答案。

二类子问题 $g(T,k,v)$:

事实上 T 只表示一个组是不够的，只表示若干组也是不够的，T需要表示一个树，其中每个叶子节点是一个组，并且每个节点（包括叶子）有一个权值 lim，表示以该点为根的子树中至多连 lim 条边。

可能上面说的不是很清晰，那么我们直接来分析转移情况。

首先考虑答案的第 v 位能否为 1，我们考虑每一个叶子表示的组，根据其第 v 位是否为 1 将其每个部分再度分裂为两个部分，分别命名为 $S00$，$S01$，$S10$，$S11$。$S00$ 指原先的第一个部分中第 v 位为 0 的子部分，$S01$ 指原先的第一个部分中第 v 位为 1 的子部分，$S10$ 指原先的第二个部分中第 v 位为 0 的子部分，$S11$ 指原先的第二个部分中第 v 位为 1 的子部分。

所以一个叶子中，至多能连出 $min(lim,min(|S00|,|S11|)+min(|S01|,|S10|))$ 条第 v 位为 1 的边，然后可以通过非常简单的树形dp来求出以每个节点为根的子树中最多能连多少条第 v 位为 1 的边。

那么根节点的dp值也已经求出，所以我们可以通过比较根节点的dp值与 k 的大小来判断答案的第 v 位是否为 1。

如果答案的第 v 位的值为 1 ，那么每个叶子节点可以被分裂为两个叶子节点，其组信息分别为 $make\_pair(S00,S11)$ 和 $make\_pair(S01,S10)$。而原先的叶子节点的 lim 值需要保持不变，然后就可以继续递归求解了。

如果答案的第 v 位的值为 0 ，那么 $S00$ 和 $S11$ 在充分连中间边后至多剩下一个部分的剩余度数大于 0 ，$S01$ 和 $S10$ 在充分连中间边后也至多剩下一个部分的剩余度数大于 0 。这两个部分可以构成一个新的组，替代原先的叶子节点。但值得注意的是，树 T 上的每个节点的 lim 值都需要减去其 dp 值，才能继续递归求解。

如果你耐心地看到了这里，并且不出意料地没有看懂，可以询问出题人skc

这题真是 skc 出的!