A. 魔法

先特判掉 $R=0$ 或 $B=0$。当且仅当序列恰好为 kR 个红球和 kB 个蓝球时是合法的。

> **证明：** 必要性显然。
>
> 充分性。$k=1$ 的时候显然。$k>1$ 的时候可以考虑将这个序列划分为 $[1,R+B],[R+B+1,2(R+B)],\dots$ 这 k 段，那么显然至少有一段满足里面的红球数 $\ge R$，至少有一段满足里面的红球数 $\le R$。
>
> 考虑长度为 $R+B$ 的区间从左端点为 1 开始往右移，那么这个区间内的红球数的变化每次只有 $+1,-1$ 和不变这三种情况，那么从红球数 $\ge R$ 移到红球数 $\le R$ 的过程中，一定存在一段恰好使得红球数 $=R$。
>
> 那么既然有存在一个区间恰好有 R 个红球和 B 个蓝球，那么就将这段区间删去，表示让其最后变成白色，然后递归到 $(k-1)R$ 个红球和 $(k-1)B$ 个蓝球的情况。归纳即可证明。

因此我们为了模拟这个过程，就可以考虑用一个栈维护，然后顺便记录栈中每个前缀的红球（或蓝球）个数，每次就考虑当前栈中的末尾 $R+B$ 个是否能够构成一个合法区间，如果可以就将其弹出。

时间复杂度 $\mathcal{O}(n)$。

B. 连通性

子任务 1/2

子任务 1 是枚举所有的图，然后模拟 Floyd 算法的过程来检验。

子任务 2 是蛮出的，我也不知道怎么做。

子任务 3/5

$m=0$ 只需要输出 $2^{\frac{n(n-1)}{2}}$。

$m=n$ 则要求每个连通块都是一个团，答案就是贝尔数的第 n 项。

正解

$m=1$ 的方法应该和正解是比较接近的，就不详细讲了。

记 $1\sim n-m$ 为黑点，剩下的 $n-m+1\sim m$ 为白点。

相当于限制每个连通块里的点对，都能找到一条除了端点以外都是黑点的路径。

全部是白点的连通块必须是一个团。如果一个连通块有黑点，那么这些黑点的导出子图必须连通。然后连通块里的每个白点都必须与一个黑点相连。因此就可以用 DP 来解决这个问题了。

记 $f(n,m)$ 表示 n,m 的对应答案，$g(n)$ 表示点数为 n 的简单无向连通图个数：

- $g(n)$ 是经典容斥，用所有简单无向图的个数减去不连通的个数，不连通的个数可以枚举 1 所在的连通块大小：

$$g(n)=2^{\frac{n(n-1)}{2}}-\sum_{i=1}^{n-1}g(i)\times 2^{\frac{(n-i)(n-i-1)}{2}}\times\binom{n-1}{i-1}$$

- 然后考虑 $f(n,m)$ 的转移，注意到不同的连通块之间是没有编号区别的，因此我们转移的时候强制编号为 n 的是在最后一个。
    - 先考虑全部都是白点的连通块：
$$f(n,m)+=\sum_{i=1}^{m} f(n-i,m-i)\binom{m-1}{i-1}$$
    - 接着再处理连通块里有黑点的情况
$$f(n,m)+=\sum_{i=1}^{m}\sum_{j=1}^{n-m} f(n-i-j,m-i)\binom{m-1}{i-1}\binom{n-m}{j}h(i,j)$$
其中 $h(i,j)=g(j)\times(2^j-1)^i\times 2^{\frac{k(k-1)}{2}}$。

时间复杂度 $\mathcal{O}(n^4)$。

C. 矩形

子任务 1

直接枚举 i,j 然后排序输出即可。时间复杂度 $\mathcal{O}(n^2\log n)$。

子任务 3

$h_i=1$ 的情况十分简单，面积为 i 的矩形有 $n-i+1$ 个。

可以考虑求前缀和然后二分 A_L，然后依次推出 $A_{L+1}\sim A_R$。时间复杂度 $\mathcal{O}(n)$。

子任务 4

$h_i=i$ 可以先二分 A_L，设 $f(s)$ 表示面积 $\le s$ 的矩形数量，判定只需要枚举矩形的高 h，计算对应的 $\frac{s}{h}$ 然后求和即可。先处理边界，令 $L'\leftarrow f(A_L)$。后面的部分则是一个经典问题。对于每个高 h，记一个值表示最小的 w_h 使得 $h\times w_h>A_L$。

接着就维护一个小根堆，储存当前所有 $>A_L$ 的 $h\times w_h$ 和对应的 h。每次取出 $h\times w_h$ 最小的堆顶，弹出堆顶，移动左端点和对应的 w_h 并维护当前的 A_L。时间复杂度 $O(n\log n+(R-L)\log n)$。

该做法提示了正解。

子任务 5

二分 A_L，然后考虑怎么求 $f(s)$。

我们可以建出序列的笛卡尔树，笛卡尔树上的每个结点，都有一个对应的区间 $[x,y]$，假设 $[x,y]$ 中的最小值为 h，位置为 p。（h 相同时将 p 更小的看成更小）

那么记 $l=p-x,r=y-p$，那么就考虑**当矩形的顶部取在 p 这个位置**时，有多少个矩形符合条件。

设 $w=\lfloor\frac{s}{h}\rfloor$，那么就要求矩形的宽 $\le w$，且必须包含 p 这个位置。我们可以容斥计算：

$$f(s)+=S(w)-S(w-l-1)-S(w-r-1)+S(w-l-r-2)$$

其中 $S(w)=\sum_{i=1}^{w} i=\begin{cases}\frac{w(w+1)}{2} & ,w>0\\ 0 & ,w\le 0\end{cases}$。

于是就可以 $\mathcal{O}(n)$ 计算 $f(s)$，然后套个二分可以通过 $R-L+1\le 3$。时间复杂度 $\mathcal{O}(n\log n)$。

该做法也提示了正解。

正解

结合子任务 $4, 5$ 的思路不难得到正解。

同样先二分得到 A_L，并处理边界，令 $L' \leftarrow f(A_L)$，只需要先输出 $L' - L + 1$ 个 A_L 即可。

然后考虑对于每个笛卡尔树上的结点 x，根据子任务 5 的算法得到三元组 (l_x, r_x, h_x)，维护出当前的 $w_x = \lfloor \frac{A_L}{h} \rfloor + 1$。

然后以 $w_x h_x$ 为键值维护小根堆，每次取出堆顶并计算对应的矩形个数，然后输出即可。

时间复杂度 $\mathcal{O}(n \log n + (R - L) \log n)$。

D. 排列

对于一个划分，逆序对乘积的期望，即为**每个划分每条线段中选两个数**，所有这些数对**都是逆序对的概率**求和。

同时我们注意到，我们可以将偶数位置排序的限制，改成偶数位置也可以任选，但是只有顺序正确的方案是合法的，最后乘上 $n!$ 即可。

$k = 1$

即求逆序对的期望个数，枚举数对分类讨论计算即可，具体不展开。

$k = n$

可以画出限制的大小关系图，是一个类似树的结构，容易推出最后的答案。

这个做法对正解也有部分提示作用，也由此可以想到下面这个引理。

引理

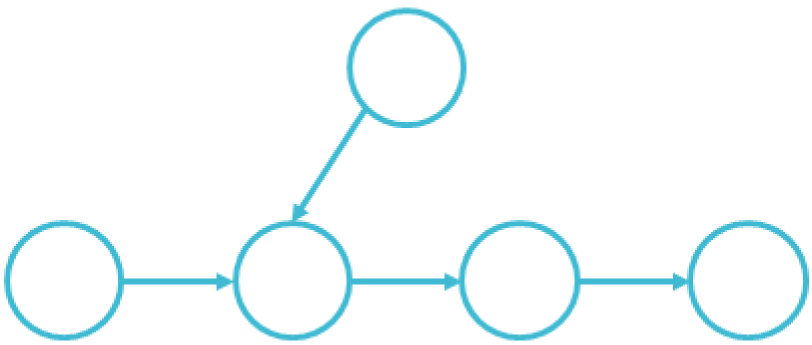
引理：一棵树给顶点随机一个排列，每个点标号均比父亲小的概率，为**所有节点子树大小倒数的乘积。**

这个结论比较显然，我们考虑对于每棵子树，当且仅当这个结点是子树中最大的才是合法的。不同结点的概率互不影响，所以直接乘起来就对了。

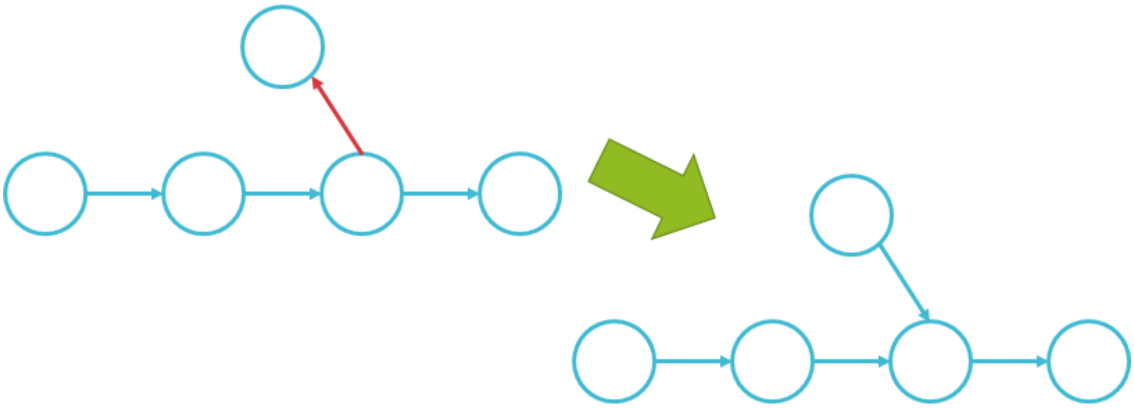
正解

所有偶数位置形成了一个链。在每段里面选择两个位置，将其都是逆序对的概率求和即为答案，考虑分类讨论算概率：

- 如果某段选的两个位置是奇数，概率直接是 $\frac{1}{2}$。
- 如果某段选的两个位置是偶数，概率为 0。
- 如果偶数位置在奇数位置前面，那么偏序关系形成的仍然是一棵树。



- 如果偶数位置在奇数位置后面，那么形成的就不是一棵树了。但是我们可以考虑容斥，用 1 减去不是逆序对的概率。



发现这棵树是从前面向后面连边的，考虑在 DP 状态中记录当前子树大小。设 $f(i,j,k)$ 表示考虑了前 $2i$ 个位置，分成了 j 段，当前子树大小为 $k+i$。

我们发现，假设最后的子树大小为 $k+i$，那么如果限制的偏序关系是一条链，那么可以发现这样的概率贡献就是 $\frac{1}{(k+i)!}$。但是实际上的限制是可能有几个点单独指向链的某些点，我们发现贡献相当于乘上一些数，**这个贡献我们就在的对应位置计算即可。**

转移分类讨论一下：

- 奇奇：$f(l,j+1,k+0)+=f(i,j,k)\frac{(l-i)(l-i-1)}{4}$
- 偶奇：$f(l,j+1,k+1)+=f(i,j,k)\sum_{t=1}^{l-i}(k+i+t)(l-i-t)$
- 奇偶：

$$f(l,j+1,k+0)+=f(i,j,k)\frac{(l-i)(l-i+1)}{2}$$
$$f(l,j+1,k+1)-=f(i,j,k)\sum_{t=1}^{l-i}(k+i+t)t$$

时间复杂度 $\mathcal{O}(n^4)$。