T1 – 棋子

解法:

在 0 号格时有 $\frac{1}{n}$ 的概率直接进入终点。
对另外 $\frac{n-1}{n}$ 概率发生的情况，可以发现从 1 号格到 $n-1$ 号格花费 1 的代价进入终点的期望是相同的，都是 $\frac{1}{n-1}$，另外 $\frac{n-2}{n-1}$ 的概率继续停留在$1\sim n-1$号格内，故在这个问题上，位于$1\sim n-1$号格子上的状态都是等价的。所以期望花费的代价为$1+\frac{n-1}{n}\times(n-1)$

T2 – 数列

解法:

20pts

$f[i][j]$ 表示 $b_i=j$ 的情况下，填入 b 序列的前 i 项的方案数。 那么 $f[i][j]$ 可以转移到 $f[i+1][k]$ 当且仅当$j\le b_i,k\le b_i,j\ne k$，假设值域为V，时间复杂度为$O(nV)$。

40pts

首先直觉告诉我们合法方案数不好求，所以正难则反考虑求不合法的情况，不合法的情况也不好求，但是我们可以比较自然地想到容斥。

也就是求 $F(i)$ 表示至少有 i 个坏点的方案数。

然后考虑动态规划，设 $f_{i,j}$ 表示前 i 项划分成 j 部分，部分内部是相同的数字，但是相邻的部分不要求不同（因为是容斥，保证至少多少个不同即可）。

显然 $F(i)=f_{n,n-i}$。 我们发现这个东西和最小值有关系，一个部分的取值方案就是最小值，这点比较显然。 转移为 $f_{i,j}=\sum_{k=1}^{i-1}f_{k-1,j-1}\times\min_{x=k}^{j}a_x$

时间复杂度为$O(n^3)$。

60pts

我们并不关心 j 的具体大小，只关心 j 的奇偶性，因为容斥系数为 $(-1)^{n-j}$，所以每取一段，就让方案数乘上 -1，我们可以忽略掉 j 这一维 $f_i=\sum_{k=1}^{i}-f_{k-1}\times\min_{x=k}^{j}a_x$ 时间复杂度为$O(n^2)$。

100pts

考虑怎么维护 $\min_{x=k}^{j}a_x$，可以维护单调栈的具体形态，利用前缀和在单调栈变化时进行 DP 转移，时间复杂度为$O(n)$。

T3 – 数表

解法:

因为一共有 $2n+1$ 个格子，所以假设有一组方案，满足每行每列均不同，那么给所有数异或一个值 $y\in[0,2^k-1]$，依然满足每行每列不同，且最终答案异或了 y。从这个角度，不管 q 取什么值，答案均相同。 因此，可以直接不管异或和的限制考虑，最终答案除以2^k即可。

填好第二行，方案数为$2^k \times (2^k-1) \times ... \times (2^k-n)$，对于第一行，我们容斥其中有 c 列和第二行相同，剩下的方案数为 $(2^k-c) \times ... \times (2^k-n+1)$。因此答案为 $\frac{1}{2^k}\sum_{i=0}^{n}(-1)^i C_n^i \times \frac{(2^k)!(2^k-i)!}{(2^k-n-1)! \times (2^k-n+1)!}$ 时间复杂度为$O(n)$

T4 – 最小环

解法:

30pts

建立一个超级源点 0，连向所有的点，边权为 0。

令$f[i][j]$表示从 0 出发走i条边到j的最短路径，答案即为 $\min_{i=1}^{n}\{\max_{j=0}^{n}\frac{f[n+1][i]-f[j][i]}{n+1-j}\}$

也可以利用状压 DP 或者二分答案后在图中找最小环（经典求平均最小环做法，但和本题关系不大，故不展开）。

80pts

因为边权是 998244353 从 0 到 $n-1$ 的幂，所以$f[i][j]$可以用一个长度为n的数组w表示，其值为 $w_i \times 998244353^i$，运算过程只会涉及单点加 1、字典序比大小以及拷贝，在最后计算答案时，可以通分后比较分子来实现，时间复杂度为$O(n^2m+n^3)$。

100pts

上述所有过程可以利用可持久化线段树进行，单点加一可以看作只修改了一个节点，字典序比较可以维护每个节点的哈希值，在线段树上二分找到第一个不同的位置，时间复杂度为$O(n^2\log n + nm\log n)$。