Day2
Solution

dy0607

October 7, 2018

1 Reverse

考虑当前在每个点时可以通过一次翻转转移到哪些点，直接BFS即可算出每个点的所需步数。然而边数会达到$O(n^2)$级别。

可以发现转移到的点一定是一段区间内的奇数或者偶数点，于是一种简单的优化方法是在BFS时开两个set维护当前有哪些奇数点和偶数点还未被BFS到，转移时直接在set上lowerbound，就不会访问已经BFS到过的点了。$O(n \log n)$

2 Silhouette

题目实际是问下面的方程组有多少非负整数解：

$$\forall i \in [1,n], \max_{j=1}^{n} x_{i,j} = A_i, \max_{j=1}^{n} x_{j,i} = B_i$$

不妨先将A, B排序，显然这样并不会有什么影响。如果A的最大值不等于B的最大值则答案是0，否则一定有解。

显然有$x_{i,j} \le \min(A_i, B_j)$. 从大到小枚举每个$A, B$中出现的值$s$，每次确定所有$\min(A_i, B_j) = s$的位置的值，我们先来考虑$s$为最大值的情况，此时我们要确定的位置构成了一个矩形。可以发现我们要解决这样一个子问题：一个$a \times b$的矩形，每个位置的值在$[0, s]$中，且每行每列的最大值均为s，需要计算有多少取值的方案。

考虑容斥，设$f(i)$为至少有i行的限制不满足条件时的方案(需要保证每一列都满足条件)，那么

$$f(i) = \binom{a}{i}(s^i \times ((s+1)^{a-i} - s^{a-i}))^b$$

方案数就是$\sum_{i=0}^{a}(-1)^i f(i)$。

s不是最大值时，我们每次要确定的位置可能是一个'L'形，仍然可以用相同的方法容斥。

由于需要快速幂，复杂度为$O(n \log n)$.

3 Seat

一个结论是，对于任意一个人，他坐下时离最近的人的距离是固定的，不随前面的人的决策而改变。这样我们可以将所有人按坐下时的距离分成若干层。另一个结论是，无论之前每一层如何决策，轮到这一层时,空区间的长度构成的可重集也是固定的。

对于最后一层特殊处理, 接下来均默认不是最后一层。对于之前的每一层，考虑在哪些空区间中央坐下可使得距离最大，其中可能会有一些长度为奇数的区间和一些长度为偶数的区

间，而对于每个人来说，坐在任意一个奇数的区间的中央的概率都是相等的，偶数同理。

那么我们只需要知道，每个人有多大的概率坐在一个奇/偶数区间的中央。考虑DP, $dp(i,j)$表示这一层已经坐下i个人之后，还有j个长度为偶数的区间的概率，转移只需考虑当前这个人坐了哪类区间即可。dp之后容易算出之前要求的概率。

区间长度为奇数时位置是固定的；考虑区间长度为偶数的情况，此时会出现两个位置可供选择，但无论选择哪一个，都会将区间划分成长度为$\frac{n}{2}$和$\frac{n}{2}-1$的两段。因此这两种选择具有对称性，我们只需要假定选择其中的一个，算出这种情况下之后的层的答案，即可对称地推出另一种情况下的答案。

瓶颈在利用对称性推答案的地方，复杂度为$O(n^2 \log n)$。