Day 1题解

陈凌峰

跳跃

- 首先考虑暴力设dp，设$f_{i,j}$表示跳跃的总长度为i，跳到的坐标位置为j的方案数
- 对于$f_{i,j}$，我们枚举跳到这个状态的上一个状态，上一个状态往左或者往右跳了k步跳到了当前状态
- 于是就有$f_{i,j} = \sum_k (f_{i-k,j-k} + f_{i-k,j+k})$
- 奇偶性的做法与直接求方案数的方法类似，设一个数组$g_{i,j}$表示跳跃的总长度为i，跳到的坐标位置为j的方案数的奇偶性
- 时间复杂度为$O(n^3)$

跳跃

- 考虑优化这个dp，设$l_{i,j} = \sum_k f_{i-k,j-k}$，$r_{i,j} = \sum_k f_{i-k,j+k}$
- 那么就有$f_{i,j} = l_{i,j} + r_{i,j}$
- $l_{i,j} = \sum_k f_{i-k,j-k} = f_{i-1,j-1} + \sum_k f_{i-k-1,j-k-1} = f_{i-1,j-1} + l_{i-1,j-1}$
- 同理$r_{i,j} = f_{i-1,j-1} + r_{i-1,j-1}$
- 那么我们便可$O(n^2)$通过$\mathrm{opt} = 1$的所有情况（注意空间限制）

跳跃

- 我们还可以从其他方向入手，把向左和向右看成网格图上从左上角的(0,0)开始向下和向右移动
- 设dp的同时最好能通过下标表示出题目中的n和m
- 设$f_{i,j}$表示向左移动（即网格图上的向下移动）的总长度为i，向右移动的总长度为j的方案数
- $\mathrm{n} = \mathrm{i} + \mathrm{j}$，$\mathrm{m} = \mathrm{j} - \mathrm{i}$
- 对于$f_{i,j}$，我们可以枚举跳到这个状态的上一个状态
- 那么就有$f_{i,j} = \sum_k (f_{i-k,j} + f_{i,j-k})$
- 可以进行前缀和优化，设$g_{i,j} = \sum_k f_{i-k,j} = g_{i-1,j} + f_{i-1,j}$，$h_{i,j} = \sum_k f_{i,j-k} = h_{i,j-1} + f_{i,j-1}$
- 由于$f_{i,j} = \sum_k (f_{i-k,j} + f_{i,j-k}) = g_{i,j} + h_{i,j}$
- 那么我们也可以通过$O(n^2)$通过opt = 1的所有情况（注意空间限制）

跳跃

- 我们还可以进一步优化空间
- 观察以下的$f_{i,j}$表格，对于i > 0，j > 0且i、j不均为1的$f_{i,j}$，其值可以转换为$2f_{i-1,j} + 2f_{i,j-1} - 3f_{i-1,j-1}$
- 这样我们只需要开一个大小为n^2的数组即可

1	1	2	4	8	16
1	2	5	12	28	64
2	5	14	37	94	232
4	12	37	106	289	760
8	28	94	289	838	2329
16	64	232	760	2329	6802

跳跃

- 考虑$opt = 2$的情况
- 观察以下的$f_{i,j}$表格，不难发现只有当$i = j = 0$或$i = j + 1$或$j = i + 1$时，$f_{i,j}$（没取模前）为奇数
- 这是为什么呢?
- 由于对于$i > 0$，$j > 0$，且i、j不均为1的$f_{i,j}$，其值可以转换为$2f_{i-1,j} + 2f_{i,j-1} - 3f_{i-1,j-1}$，所以$f_{i,j}$的奇偶性与$f_{i-1,j-1}$的奇偶性相同

1	1	2	4	8	16
1	2	5	12	28	64
2	5	14	37	94	232
4	12	37	106	289	760
8	28	94	289	838	2329
16	64	232	760	2329	6802

好的排列

- 对于第一档部分分，我们采取直接爆搜的方法即可

- 对于第二档部分分，$A_i \neq -1$，我们可以直接dp，设$f_{i,j}$表示当前走到(i, j)，并且没有经过任何一个障碍点的方案数，直接$O(n^2)$递推即可

好的排列

- 我们考虑如果对于一个i，$A_i \neq -1$，我们只需要不经过就可以了
- 我们设满足$A_i \neq -1$的i共有K个,以下的所有讨论都基于不经过这些已知的障碍点的前提
- 比较难处理的是对于$A_i = -1$的情况,因为P是一个排列,对于一个i满足$A_i = -1$，我们不可能记录下我们当前让哪个点作为障碍点，这个状态实在太多了
- 我们考虑容斥的做法(子集反演)，令X表示合法的路径经过的点的集合
- 令h(S)表示对于所有恰好经过的障碍点集合为S，所有的好的排列P的和，即$h(S) = \sum_X \sum_P [X \cap P = S]$，所求的即$h(\emptyset)$
- 令$g(S) = \sum_{S \subseteq T} h(T)$，则有$h(\emptyset) = \sum_T (-1)^{|T|} g(T)$

好的排列

- 考虑g(S)怎么求
- $g(S)=\sum_{S\subseteq T}\sum_{X}\sum_{P}[X\cap P=T]=\sum_{X}[S\subseteq X]\sum_{P}[S\subseteq P]=\sum_{X}[S\subseteq X]\times(n-K-|S|)!$
- 所以我们仅需dp求出$\sum_{X}[S\subseteq X]$，然后乘以(n − K − |S|)!即可
- 不妨令f_{i,j,k,o_1,o_2}表示当前走到(i, j)，集合S的大小为k，o_1,o_2表示当前行/列上是否有已经钦定的障碍点（即自己选的障碍点）的方案数
- 时间复杂度为$O(n^3)$

网格

- 首先可以想到一个$O(n^4)$的做法，设$f_{x1,y1,x2,y2}$表示从网格(x1, y1)走到(x2, y2)所需要消耗的最小体力值，这样便能轻松拿到40分了

- 网格图，容易联想到对其中一维进行分治
- 我们将询问离线下来，对于x1、x2所在的方向进行分治，设当前区间为[l, r]，$\mathrm{mid} = \frac{l+r}{2}$，只处理$l \le x1 \le x2 \le r$的询问
- 对于$x2 < mid$的情况，传递到区间$[\mathrm{l}, \mathrm{mid} - 1]$去处理
- 对于$x1 > mid$的情况，传递到区间$[\mathrm{mid} + 1, \mathrm{r}]$去处理
- 只需要考虑$x1 \le mid \le x2$的情况

网格

- 对于$x1 \le mid \le x2$的情况，我们在y1、y2所在的方向去枚举一个点$(\mathrm{mid}, \mathrm{i})$，我们从$(\mathrm{x1}, \mathrm{y1})$走到$(\mathrm{mid}, \mathrm{i})$，再走到$(\mathrm{x2}, \mathrm{y2})$
- 我们可以设$f_{i,j,k}$表示从(i, j)正着走到$(\mathrm{mid}, \mathrm{k})$所需要消耗的体力值，设$g_{i,j,k}$表示从$(\mathrm{i}, \mathrm{j})$倒着走到$(\mathrm{mid}, \mathrm{k})$所需要消耗的体力值
- 那从$(\mathrm{x1}, \mathrm{y1})$走到$(\mathrm{x2}, \mathrm{y2})$的答案即为$min(f_{x1,y1,i} + g_{x2,y2,i})$
- 时间复杂度$O(nm^2 log\ n + qm)$