ONTAK 2010 day2 解题报告

雅礼中学 刘剑成

1 试题来源

ONTAK 2010 day2 - Keyboard
ONTAK 2010 day2 - Garden
ONTAK 2010 day2 - Matchings

2 Keyboard

2.1 试题大意

有一个$n \times m$的棋盘，n和m均为奇数。棋盘的每个格子都写上了一个小写字母，并且棋盘被$\frac{nm-1}{2}$个1×2的多米诺骨牌所覆盖，只有左上角的格子没有被覆盖。

你有一种操作，每次可以选择一个骨牌进行移动，要求横向的骨牌只能横向移动，纵向的骨牌只能纵向移动，且在移动之后不能有两个骨牌重合。在每次移动之后，你都可以将未被覆盖的格子涂黑。

移动需要1个单位的时间，涂黑不计时间。

要求棋盘上所有元音字母（a,e,i,o,u,y）所在的格子给涂黑，问最少需要多少单位的时间。

$1 \le n, m < 70$。

时间限制：2sec
空间限制：32MB

2.2 算法介绍

由于整个棋盘上只有一个空格，则我们每移动一次，必然是将周围四个方向上的一块多米诺移动至当前空格，即将纵坐标或横坐标加减2。

定理2.1. *假设当前没被覆盖的格子为(x,y)。若不连续移动一块多米诺2次，则不可能有一种移动方法，使得移动了至少一次，且未被覆盖的格子回到了(x,y)。*

使用反证法，假设存在一种移动方法使得没有被覆盖的格子最终回到了(x,y)。若该条路径在除了(x,y) 以外的点(x',y')自交了，则选择从(x',y')开始至(x',y')结束的路径进行证明。

假设未被覆盖的点走过的路径为：

$$(x,y) \to (x_1,y_1) \to (x_2,y_2) \to ... \to (x_k,y_k) \to (x,y)$$

则从(x_1,y_1)开始直到(x_k,y_k)，每个点上所放置的多米诺放置方向必然朝向之前一个点，且相邻两个点必然为横坐标或纵坐标相差2，否则不能从前一个点通向当前点。又由于(x,y)初始为空，在移动之后必然被朝向(x_1,y_1)的多米诺所覆盖，且$(x_1,y_1) \neq (x_k,y_k)$，所以必然不能从(x_k,y_k)移动至(x,y)。**定理**2.1证毕。

由**定理**2.1可得空格可以到达的所有位置构成了一棵树，我们只需要判断所有元音是否在树中即可判断该问题是否有解。

接下来，原问题被转化为：从树根出发，每次可以沿着一条树边行走，问遍历一棵树中所有被标记的点最少需要多少步。这个问题我们可以转化为在最后返回原点最少需要多少步减去最后返回的长度。

对于第一部分，我们不断地去掉整棵树中没有被标记的叶子节点，直到最后所有的叶子节点均为被标记的节点，遍历所有节点所需要的代价即为答案——即$2(T-1)$[1]。

对于第二部分，可以发现最后返回的长度最长的即为最深的被标记的节点，所以答案即为该节点的深度。

至此，问题解决。

时间复杂度：$O(nm)$

空间复杂度：$O(nm)$

[1]T指在去掉多余的节点之后剩余的树的大小

3 Garden

3.1 试题大意

给出n个点，要求从这n个点中选出四个点，使得选出的点构成与坐标轴平行的正方形，问一共有多少种不同的选法。

$1 \le n \le 10^5$，坐标绝对值范围在10^6以内。

时间限制：3sec/12sec

空间限制：64MB

3.2 算法介绍

首先我们从暴力开始思考：枚举两个点，通过hash判断另外两个点是否存在。时间复杂度为$O(n^2)$，很明显不能被接受。

考虑暴力的一个优化：每次枚举的时候仅枚举当前行的点。虽然在优化了之后时间复杂度仍然没有变，但是为我们之后的解题提供了一个很好的思路。

再次考虑怎样的数据才能将优化后的暴力卡至$O(n^2)$，答案很简单，即所有的点均在一条链上。可以发现，当这个时候我们如果采取另一个方法来解决，也许可以避免时间复杂度太差的情况。

假设当整行中所需要枚举的点个数不超过S时，此时我们使用优化后的暴力，时间复杂度为$O(nS)$。这样所有至少一条边所在的行中点数不超过S的正方形均被计入了答案（注：当两条边均不超过S时不要重复计数）。

现在剩下的只有当正方形的上下两边所在的行中点的个数均超过S的情况。可以发现，这样的行是不超过$\lfloor \frac{n}{S} \rfloor$个的，所以我们暴力枚举每个点之后，再次枚举下边界在哪一行，通过hash判断剩余的3个点是否存在即可，这样的复杂度为$O(n \cdot \frac{n}{S})$。

算法的总时间复杂度为$O(nS + \frac{n^2}{S})$，可以发现当S取$O(\sqrt{n})$时时间复杂度最优，为$O(n\sqrt{n})$。

时间复杂度：$O(n\sqrt{n})$

空间复杂度：$O(n)$

4 Matchings

4.1 试题大意

给出一棵大小为n的树，现在你要选出若干条边，满足这些边之间没有公共端点。问最多选出多少条边，且在选出最多边的前提下有多少种不同的选择方案，答案对m取模后输出。

$1 \le n \le 1.5 \times 10^6$，$1 \le m \le 10^9$。

时间限制：2sec/8sec

空间限制：32MB

4.2 算法介绍

对于第一问，有一个简单的贪心思路：

当一个叶子与另一个点相连，可以直接选中这两个点之间所连的边，接下来去掉这两个点以及与它们相连的边，直到最后图中没有边为止。

对于第二问，可以使用动态规划进行处理，我们给这棵树任意定一个根，则可设出状态：$F_{i,j,k}$代表i节点所对应子树中边的选择情况已经确定，且选择了k条边，点i的选择情况为j时的方案数有多少。

然而这样设出的状态是$O(n^2)$的，而n在最大时有1.5×10^6，很显然是不可能通过的。可以发现，当k不是当前i, j所对应的最大的值时，它必然不是最优的。所以我们只需要记录当k为最大时所构成的值即可将状态优化至$O(n)$。

从时间复杂度上来说，上述算法所需要的时间为$O(n)$，本题已经被完美的解决，但问题在于本题的空间限制只有32MB，对于$n = 1.5 \times 10^6$的数据，32MB仅能开5个int数组，所以需要更加优化在处理时的细节。

我们从最主要的部分——动态规划的部分开始思考，首先对于每一个i，我们需要记录当k为0或1时两个值。而且转移时需要记录k为0或1时当前的最大匹配为多少，这里需要大小为$4n$的空间。剩下的只有大小为n的空间用来记录树的形态，而只用1 倍的空间来记录树的形态中，对于此题一个良好的方法是记录父亲，又由于需要按照一定的顺序来遍历，且没有剩余的空间来记录，所以我们在这之前需要按照遍历的顺序重标号一次。

注意在读入边的时候需要保留双向边，即边集的大小为$2n$，而邻接表存边

所需要的空间至少为$2n + m$[2]，剩余的空间最多开出一个大小为n的数组，已经不足以支持之后的Bfs，所以需要用到前向星来存边。

前向星所需要的空间为$3n$，即将所有的边按照连入的点标号排序之后，用一个大小为n数组用来存从同一个点连出去所有边的起始位置，接下来用一个大小为$2n$的数组来记录每条边连向了哪个点。这样虽然可以节省比较多的空间，但是需要花费$O(n \log n)$的时间将边排序，相当于我们用时间来换取了空间。

时间复杂度：$O(n \log n)$
空间复杂度：$O(n)$

[2]此处m指总边数，即$2n$