

## CodeM2018 初赛 A 轮题解

Remote :

Idea :

按照题意模拟按键的过程，连续两个按键之间的操作数等于它们的曼哈顿距离，累加即可。

时间复杂度为  $O(\text{字符串长度})$ 。

Solution :

<https://paste.ubuntu.com/p/BhVNYShQQc/>

Chess :

Idea :

为了方便，不妨设坐标从 0 开始，考虑从  $j$  走到  $i$ ，如果  $i \leq j$  代价为  $j-i$ ，否则为  $j+i$ ，所以只要我们匹配一对  $(i, j)$ ，使得  $i < j$ ，就相当于减少了  $2i$  的花费，于是我们每次贪心地选最大的可以被匹配的  $i$  匹配，从大往小 for 一遍即可。

时间复杂度为  $O(n)$ 。

Solution :

<https://paste.ubuntu.com/p/qgyX8Cz4M3/>

Tree :

Idea :

除了从  $S$  到  $T$  的主路径上的边经过奇数次，其他的边都要经过偶数次（从主路径上出去一次，对应的就要回来一次）。那么答案等于 所有路径经过偶数次的代价 - 主路径上的边经过偶数次的代价 + 主路径上的边经过奇数次的代价。第一项是一个固定的值，可以预处理，后面两项可以通过求  $lca$ +前缀和的方法来维护。此题得解。

时间复杂度为  $O(n+m \log n)$ 。

Solution :

<https://paste.ubuntu.com/p/4JSyPx48rH/>

Maze :

Idea :

对原迷宫做多源的 BFS，每当有两个不同的源搜索到同一个单元时则找到一条图  $G$  中的边。容易证明 Kruskal 算法中存在一种解，使得该解中的所有边都会按这些边的长度从小到大的顺序被找到。

时间复杂度为  $O(nm)$ 。

Solution :

<https://paste.ubuntu.com/p/DnSDFbwjPT/>

Tower :

Idea :

对于每个凸包, 贡献为 $2^{\text{内部点个数}}$ 。所以我们枚举凸包左下角的点, 然后 dp 凸包上的顶点即可。具体做法就是按犄角序排序, 然后  $dp[i][j]$  表示上一条边为  $i \rightarrow j$ , 枚举下一个点, 判断叉积以及有没有跳过关键点即可。

时间复杂度为  $O(n^4)$ 。可以优化到  $O(n^2 \log n)$ 。

Solution :

<https://paste.ubuntu.com/p/4q6zFHKndN/>

Fraction :

Idea :

我们要在一个无限小数  $a/b$  后面找到某个子串。如果在第  $k$  位出现, 那么我们可以认为是  $a \cdot 10^k / b$  的小数点后开始的位。

比如

$1234/4567=0.270199255528\dots$      $1234 \cdot 10^3/4567=270.199255528\dots$

而  $1234 \cdot 10^3 \bmod 4567=910$ ,  $910/4567=0.199255528\dots$

所以对于一个串比如 1992 就可以得到  $0.1992 \cdot b \leq a \cdot 10^k \bmod b < 0.1993 \cdot b$ 。

所以对于一个串  $S$  实际上得到了一个  $a \cdot 10^k \bmod b$  的一个范围。

如果范围过大, 那么我们可以直接枚举  $k$ , 否则可以枚举这里的数字使用 BSGS 解决。

如果范围为  $T$ , 那么在  $b/T$  级别左右就可以枚举到, 否则把 BSGS 的块设为  $G=\sqrt{bT}$  大小, 那么 BSGS 的时间复杂度为  $G+bT/G=O(\sqrt{bT})$ , 所以总的时间复杂度约为  $O(b^{\{2/3\}})$ 。

Solution :

<https://paste.ubuntu.com/p/vnncY3BJfk/>