# NOIp信心赛

by Qiaoranliqu

2016 年 10 月 19 日

# 题目来源

- *TC Div*1 *Easy*.

# 考察算法

- 枚举,贪心.

# 算法一

- 直接按照题目要求枚举.复杂度$O(N^2 * M^2 * K)$.

# 算法一

- 直接按照题目要求枚举.复杂度$O(N^2 * M^2 * K)$.
- 期望得分$50 - -70$.

# 算法二

- 考虑任意一个点,和这个点的控制范围有交的点的个数至多为$K^2$.所以只需要把前$K^2+1$的点找出来,然后每个点和这些点算一下最大值即可.

# 算法二

- 考虑任意一个点,和这个点的控制范围有交的点的个数至多为$K^2$.所以只需要把前$K^2+1$的点找出来,然后每个点和这些点算一下最大值即可.
- 复杂度$O(N*M*K^3)$.

# 算法二

- 考虑任意一个点,和这个点的控制范围有交的点的个数至多为$K^2$.所以只需要把前$K^2+1$的点找出来,然后每个点和这些点算一下最大值即可.
- 复杂度$O(N*M*K^3)$.
- 期望得分100.

# 题目来源

- *TC SRM Div1 Medium.*

# 考察算法

- *DP*.

# 算法一

- 枚举每种情况,然后做前缀和进行判断.复杂度$O(2^N * N)$.

# 算法一

- 枚举每种情况,然后做前缀和进行判断.复杂度$O(2^N * N)$.
- 期望得分30.

# K=0

- 找出所有被覆盖的位置,其他位置就可以随便放了.

# K=0

- 找出所有被覆盖的位置,其他位置就可以随便放了.
- 期望得分10.

# K=1

- 考虑从左往右开始放,现在要放$i$.

# K=1

- 考虑从左往右开始放,现在要放$i$.
- 如果$i$不放,那么继续放$i+1$.

# K=1

- 考虑从左往右开始放,现在要放$i$.
- 如果$i$不放,那么继续放$i+1$.
- 如果$i$放了,那么所有包含$i$的区间的并的位置都不能放.

# K=1

- 考虑从左往右开始放,现在要放$i$.
- 如果$i$不放,那么继续放$i+1$.
- 如果$i$放了,那么所有包含$i$的区间的并的位置都不能放.
- 注意到这些位置一定是一段连续的区间,而且假设和前面放的冲突的话,在前面就会跳过这个地方的选择.

# K=1

- 考虑从左往右开始放,现在要放$i$.
- 如果$i$不放,那么继续放$i+1$.
- 如果$i$放了,那么所有包含$i$的区间的并的位置都不能放.
- 注意到这些位置一定是一段连续的区间,而且假设和前面放的冲突的话,在前面就会跳过这个地方的选择.
- 于是就可以$dp$了,令$f_i$表示现在已经放完$i$的方案.预处理出包含每个点的区间中最右边的端点$Max_i$就可以转移了.

# K=1

- 考虑从左往右开始放,现在要放$i$.
- 如果$i$不放,那么继续放$i+1$.
- 如果$i$放了,那么所有包含$i$的区间的并的位置都不能放.
- 注意到这些位置一定是一段连续的区间,而且假设和前面放的冲突的话,在前面就会跳过这个地方的选择.
- 于是就可以$dp$了,令$f_i$表示现在已经放完$i$的方案.预处理出包含每个点的区间中最右边的端点$Max_i$就可以转移了.
- 复杂度$O(NM)$.

# K=2

- 由$K=1$得到启发,对上一个做法进行小改动.

# K=2

- 由$K=1$得到启发,对上一个做法进行小改动.
- 令$f_{i,j}$为已经做完$i$,当前所有已经包含一个1的区间中最右边的端点是$j$的方案,转移与算法三中的$dp$类似.

# K=2

- 由$K=1$得到启发,对上一个做法进行小改动.
- 令$f_{i,j}$为已经做完$i$,当前所有已经包含一个1的区间中最右边的端点是$j$的方案,转移与算法三中的$dp$类似.
- 复杂度$O(N(N+M))$.

# 题目来源

- 洛谷9月月赛C.

# 考察内容

- 观察性质,*STL*,区间求交/并.

# 算法一

- 把路径经过的点都记下来,然后枚举一下即可,用STL优化复杂度.

# 算法一

- 把路径经过的点都记下来,然后枚举一下即可,用$STL$优化复杂度.
- 复杂度$O(NK \log NK)$.

# 算法一

- 把路径经过的点都记下来,然后枚举一下即可,用$STL$优化复杂度.
- 复杂度$O(NK \log NK)$.
- 期望得分30.

# 算法二

- 观察机器人行走的过程,发现路径相当于一个图形每次向某个方向平移.

# 算法二

- 观察机器人行走的过程,发现路径相当于一个图形每次向某个方向平移.
- 现场赛中有人利用随机数据通过这题.利用的性质是当$N$很小或随机时,和某次画出来的图形有交的图形数量会比较少,然后分情况搞搞就好了.

# 算法二

- 观察机器人行走的过程,发现路径相当于一个图形每次向某个方向平移.
- 现场赛中有人利用随机数据通过这题.利用的性质是当$N$很小或随机时,和某次画出来的图形有交的图形数量会比较少,然后分情况搞搞就好了.
- 复杂度$O(N^2)$.

# 算法二

- 观察机器人行走的过程,发现路径相当于一个图形每次向某个方向平移.
- 现场赛中有人利用随机数据通过这题.利用的性质是当$N$很小或随机时,和某次画出来的图形有交的图形数量会比较少,然后分情况搞搞就好了.
- 复杂度$O(N^2)$.
- 期望得分70.

# 算法三

- 不妨考虑更一般的情况,令第一次走完指令后机器人停在$(x, y)(x > 0, y > 0)$.

# 算法三

- 不妨考虑更一般的情况,令第一次走完指令后机器人停在$(x, y)(x > 0, y > 0)$.
- 则对于第一个图形中的每个点$(A_i, B_i)$,都有$K$个点$(A_i + j * x, B_i + j * y)$,其中$0 \leq j < K$.

# 算法三

- 不妨考虑更一般的情况,令第一次走完指令后机器人停在$(x, y)(x > 0, y > 0)$.
- 则对于第一个图形中的每个点$(A_i, B_i)$,都有$K$个点$(A_i + j * x, B_i + j * y)$,其中$0 \leq j < K$.
- 现在我们将所有通过平移$(x * k, y * k)$能够相互到达的点分为一组,即将它们放在$(a, b)(0 <= a < x)$中.

# 算法三

- 不妨考虑更一般的情况,令第一次走完指令后机器人停在$(x, y)(x > 0, y > 0)$.
- 则对于第一个图形中的每个点$(A_i, B_i)$,都有$K$个点$(A_i + j * x, B_i + j * y)$,其中$0 \leq j < K$.
- 现在我们将所有通过平移$(x * k, y * k)$能够相互到达的点分为一组,即将它们放在$(a, b)(0 <= a < x)$中.
- 可以发现这些点的横坐标都相差$x$的倍数.

# 算法三

- 不妨考虑更一般的情况,令第一次走完指令后机器人停在$(x, y)(x > 0, y > 0)$.
- 则对于第一个图形中的每个点$(A_i, B_i)$,都有$K$个点$(A_i + j * x, B_i + j * y)$,其中$0 \leq j < K$.
- 现在我们将所有通过平移$(x * k, y * k)$能够相互到达的点分为一组,即将它们放在$(a, b)(0 <= a < x)$中.
- 可以发现这些点的横坐标都相差$x$的倍数.
- 枚举每组$(a, b)$作为答案正方形的左下角,那么可能成为另外三个角的点,都在$(a + 1, b), (a, b + 1), (a + 1, b + 1)$所属于的集合中.

# 算法三

- 也就是说我们要对这四个点的点集求一个在横坐标上的交($(a+1)$的集合要左移一位).

# 算法三

- 也就是说我们要对这四个点的点集求一个在横坐标上的交($(a+1)$的集合要左移一位).
- 观察每个集合,容易发现每个点集都是一些连续相差$x$的点的并.

# 算法三

- 也就是说我们要对这四个点的点集求一个在横坐标上的交($(a+1)$的集合要左移一位).
- 观察每个集合,容易发现每个点集都是一些连续相差$x$的点的并.
- 考虑到每个集合的坐标差都是$x$的倍数,可以直接当成是连续的线段,然后最后再除$x$就行了.

# 算法三

- 也就是说我们要对这四个点的点集求一个在横坐标上的交($(a+1)$的集合要左移一位).
- 观察每个集合,容易发现每个点集都是一些连续相差$x$的点的并.
- 考虑到每个集合的坐标差都是$x$的倍数,可以直接当成是连续的线段,然后最后再除$x$就行了.
- 现在要做的就是对一些线段先分别求个并,然后4个集合再求个交就行了,这个可以非常轻松地在$O(N \log N)$的时间内完成,而每个集合在成为4个角的时候都被计算了一次,所以总的复杂度就是$O(N \log N)$.