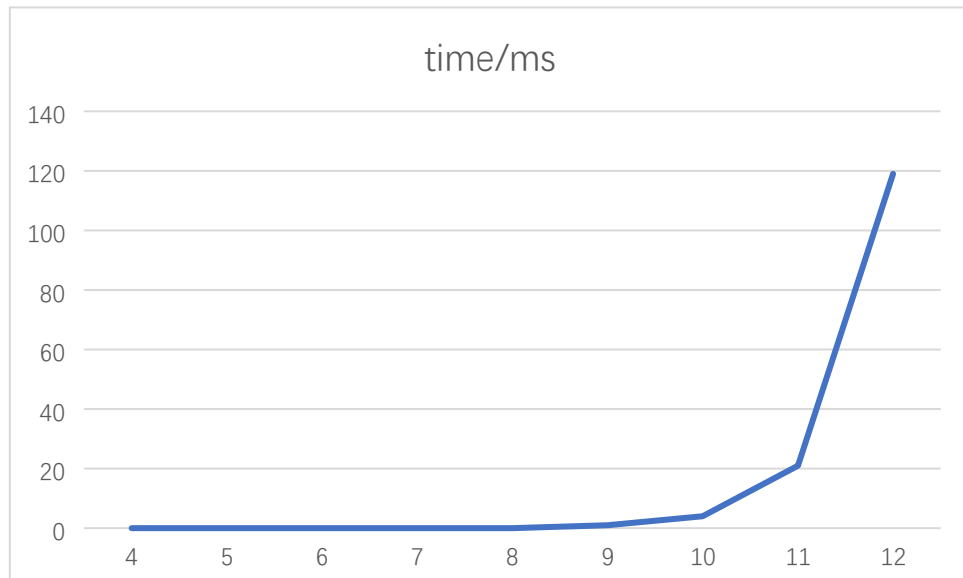


本程序利用回溯法，通过递归调用 `n_queens()` 函数实现落子。本程序有两个全局变量，`location[]` 数组负责记录第 `n` 行的棋子所在的列数，`number` 负责记录棋盘大小。`n_queens(int now,int all)` 有两个输入，`now` 记录当前落子行数，`all` 记录棋盘大小。利用 `is_safe()` 循环检查棋盘的每一列是否可落子，如果可以则落子，并调用下一层函数在下一行落子。当 `now>all` 时说明已经找到解，调用 `print()` 将其输出。以下是屏蔽掉输出后程序用时随 `n` 增长的图像。



可见程序用时随棋盘尺寸增长得十分快速。此程序的时间复杂度为 $O(N^N)$ ，需要进一步优化。

经过优化后的程序，修改了判断棋盘位置的逻辑，引入数组记录列、对角线上格子的安全性。如果不安全，则直接切换到下一个安全的位置，减少了在同一错误列反复试错的次数。此为优化后的速度。

