基于连通性状态压缩的动态规划问题

长沙市雅礼中学　陈丹琦

Email : skyfish_cdq@163.com

引入

状态压缩动态规划

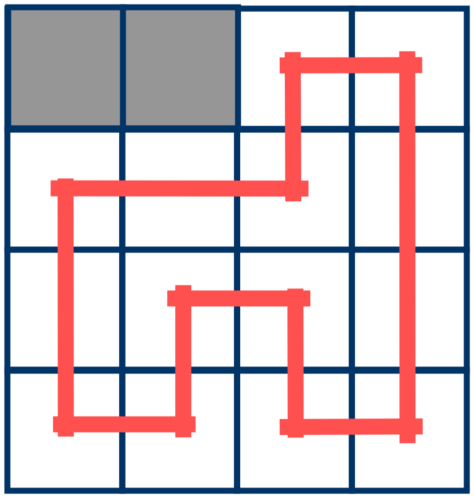
- 我的论文针对其中的一类问题进行探讨和研究——状态中需要记录若干个元素之间的**连通**情况，称之为基于连通性状态压缩的动态规划问题

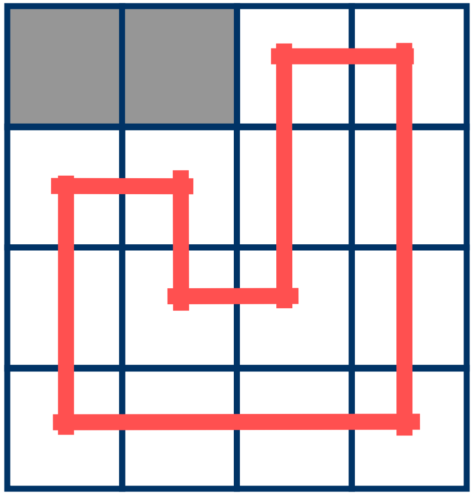
状态总数为**指数级**

【例】 Formula 1 (Ural1519)

- 一个 $m*n$ 的棋盘
- 有的格子存在障碍
- 求经过所有非障碍格子的哈密顿回路个数

$m, n \le 12$





初步分析

- **问题特点：**
  - **数据规模小** $m, n \leq 12$
  - **棋盘模型**

~~搜索？~~ $O((mn)!)$

划分阶段：从上到下，从左到右逐格递推

基本概念：插头，轮廓线

状态压缩！√

基本概念

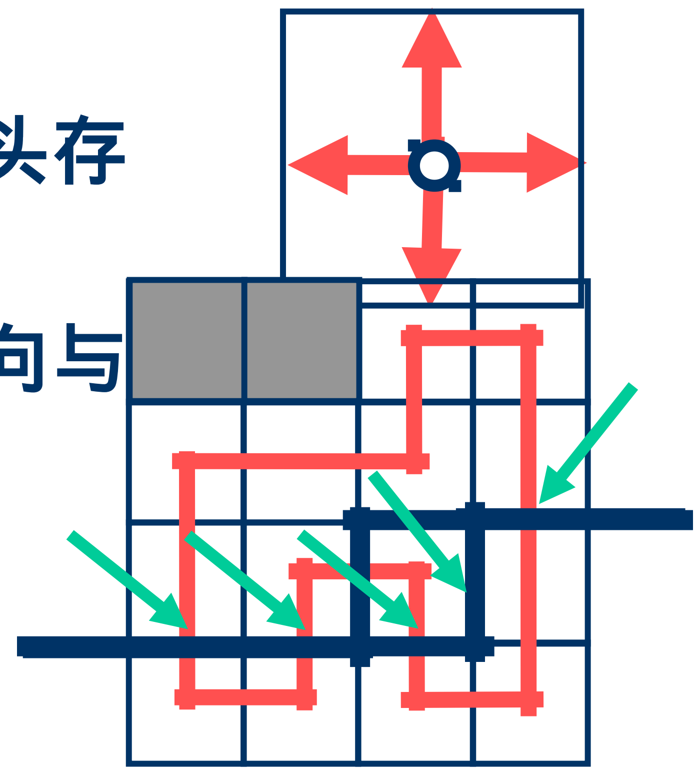
- 插头

  一个格子某个方向的插头存在表示这个格子在这个方向与相邻格子相连

- 轮廓线

  已决策格子和未决策格子的分界线

  轮廓线上方与其相连的有 $n+1$ 个插头，包括 n 个下插头和 1 个右插头．

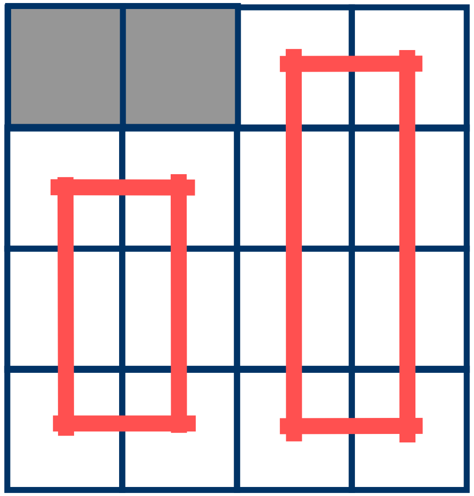


初步分析

- **问题特点：**
  - **数据规模小**
  - **棋盘模型**

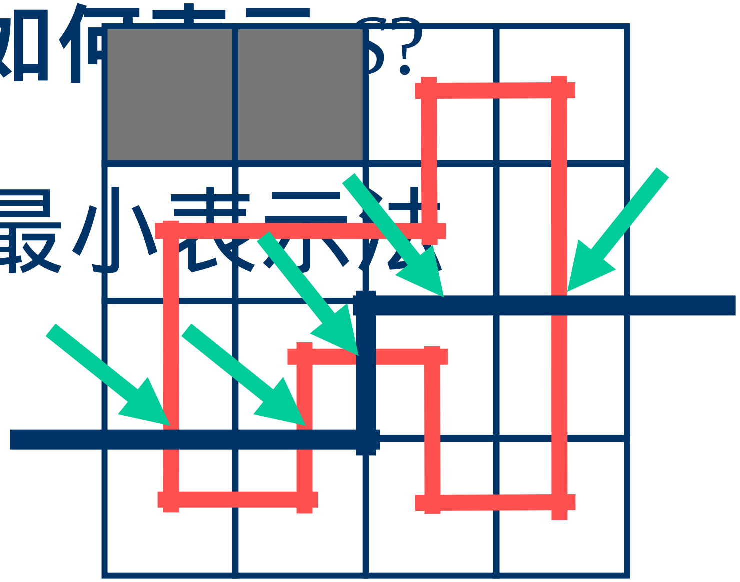
    每个插头是否存在
  - **所有的非障碍格子连通**

    插头之间的连通性！



确立状态

- 设 $f(i, j, S)$ 表示转移完 (i, j) ，轮廓线上从左到右 $m+1$ 个插头是否存在以及它们的连通性为 S 的方案总数。
- 无插头标记0，有插头标记为正整数
- 连通的插头标记相同的数字
- 从左到右依次标记
- 如何表示 S?
- 最小表示法



1	2	2	0	1

$f(3,2,\{1,2,2,0,1\})$

状态转移

- 考虑每个格子的状态，根据上一个状态 $O(n)$ 扫描计算出新的最小表示状态．
- 对于 $m = n = 12$ 的无障碍棋盘的极限数据，扩展状态总数为 1333113，问题已经基本解决．
- 本题为一个棋盘模型的**简单回路**问题．

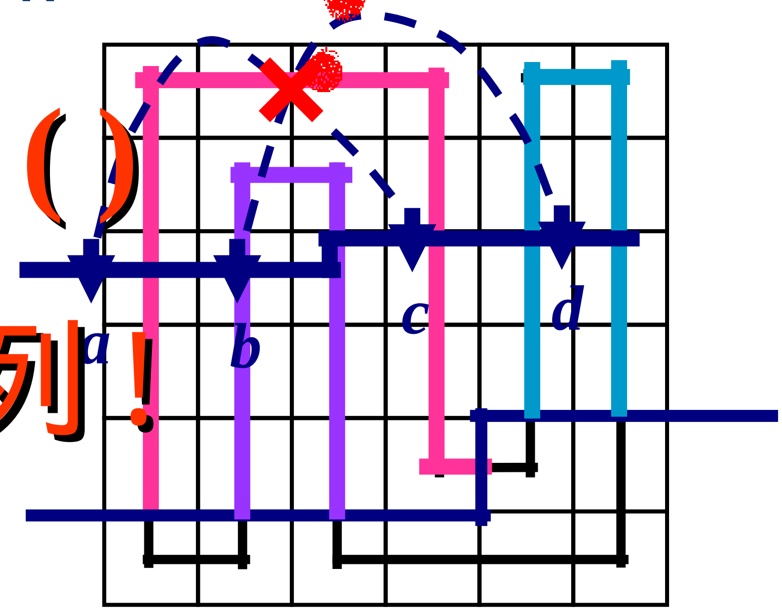
  针对问题的特殊性，是否有更好的方法呢？

进一步分析

- **每个非障碍格子恰好有** 2 **个插头**
- **轮廓线以上由若干条互不相交的路径构成**
- **从左到右一定不会出现** 4 **个插头** *a*, *b*, *c*, *d*
- **每条路径的两端对应两个插头**
- *a* , *c* **匹配**，*b*, *d* **匹配**

插头不会交叉
插头两两匹配

( ) (( )) ( )

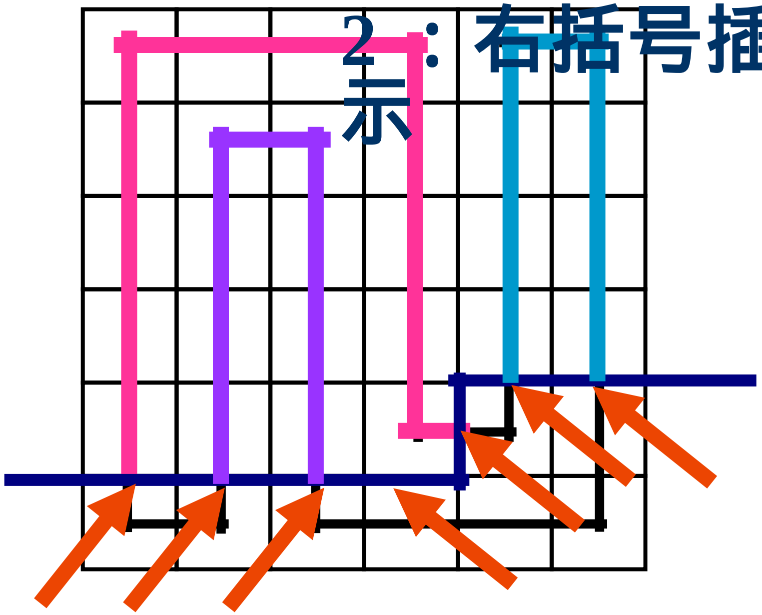


括号序列！

括号表示法

3 进制

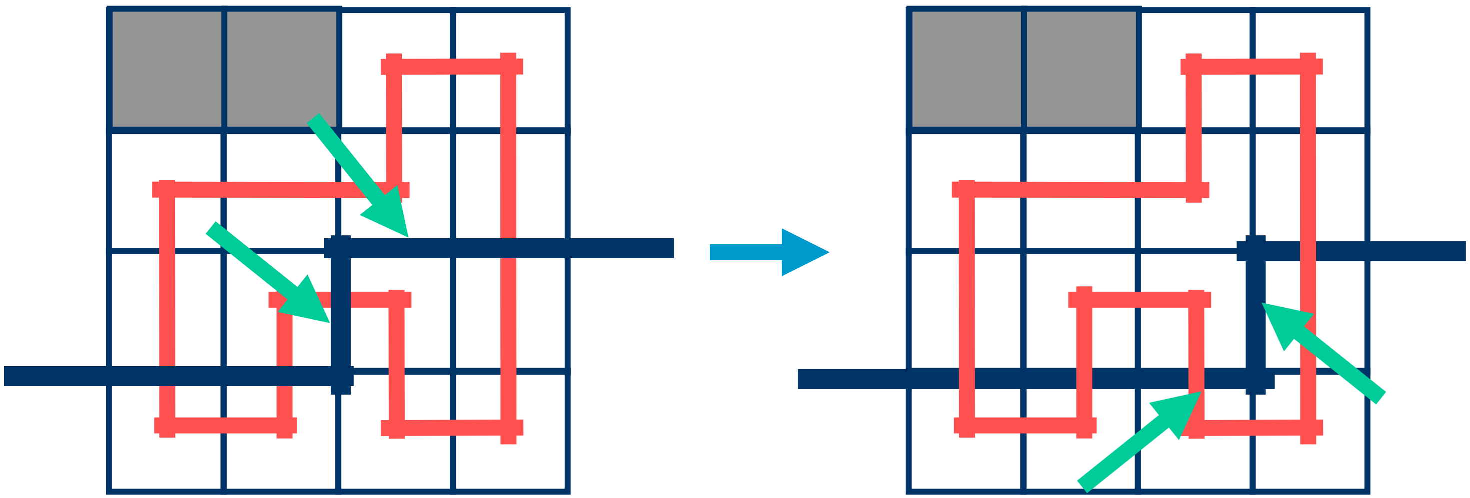
- 0：无插头状态，用 # 表示
- 1：左括号插头，用（ 表示
- 2：右括号插头，用 ）表示



（（）#）（）

$(1\ 1\ 2\ 0\ 2\ 1\ 2)_3$

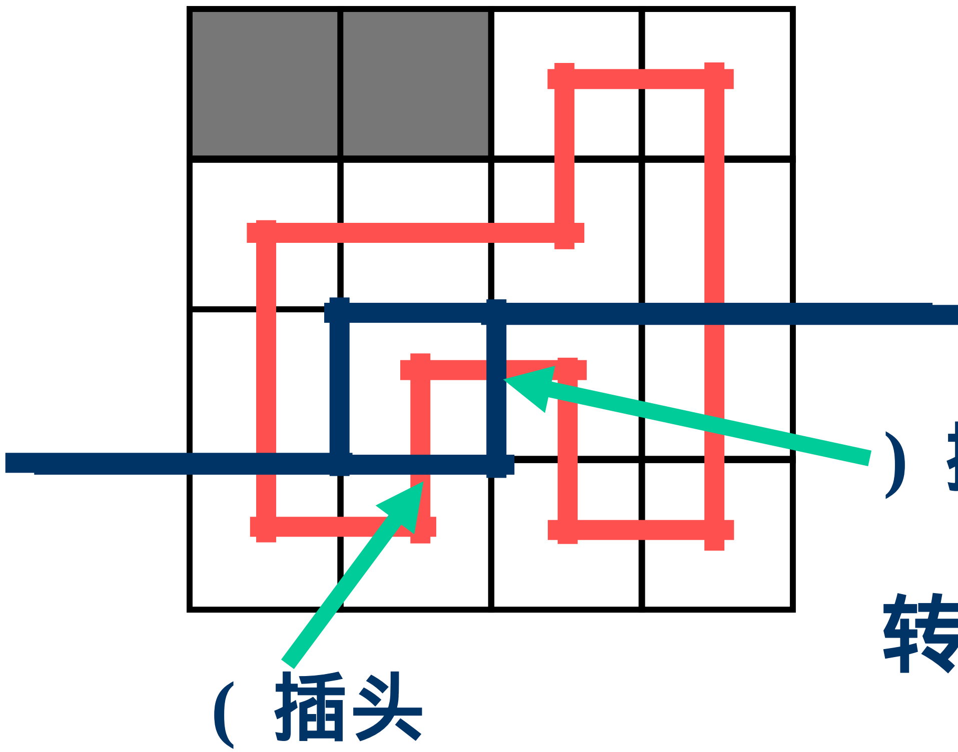
状态的转移



- 每次转移相当于轮廓线上当前决策格子的左插头改成下插头，上插头改成右插头的状态．

Case 1

没有上插头和左插头，有下插头和右插头，相当于构成一个新的连通块.



(# # #)

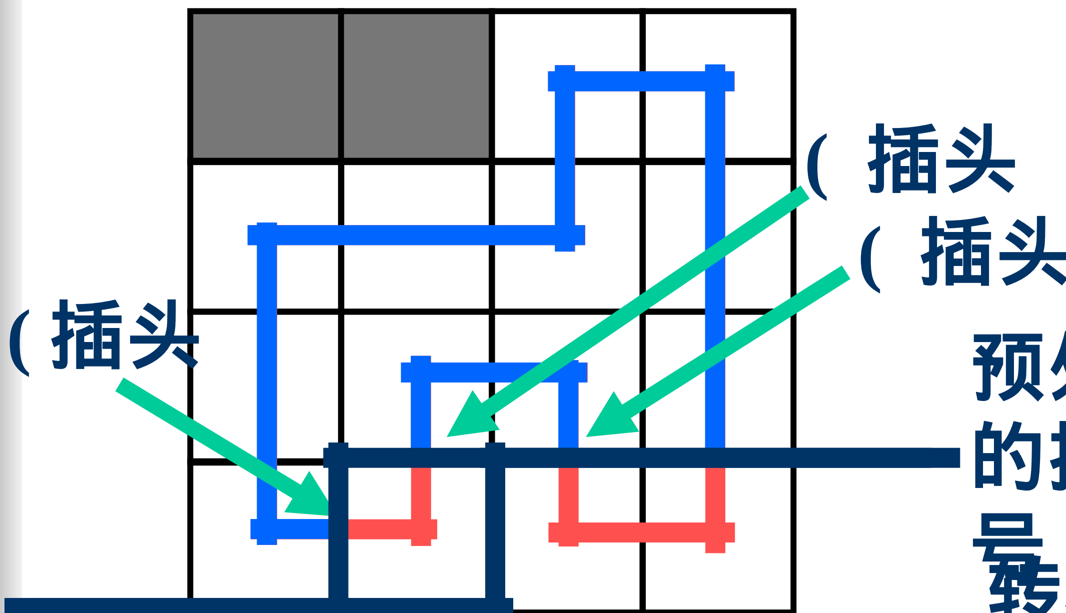
(( ) #)

转移时间： $O(1)$

Case 2

有上插头和左插头，这种情况下相当于合并两个连通分量

Case 2.1 **上插头和左插头均为 (插头**



\# (( ))

↓ ↓ ↓

(# # ( )

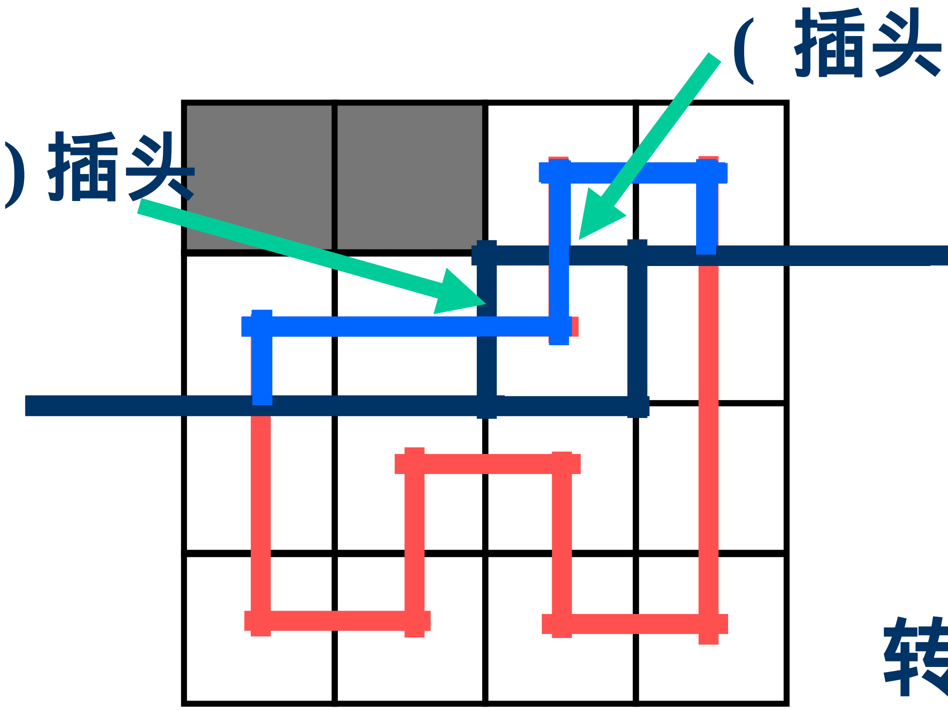
预处理每个状态每的括号所匹配的括号

转移时间：$O(1)$

Case 2

有上插头和左插头

Case 2.2 **左插头为) 插头，上插头为 (插头**



(#) ( )

↓ ↓

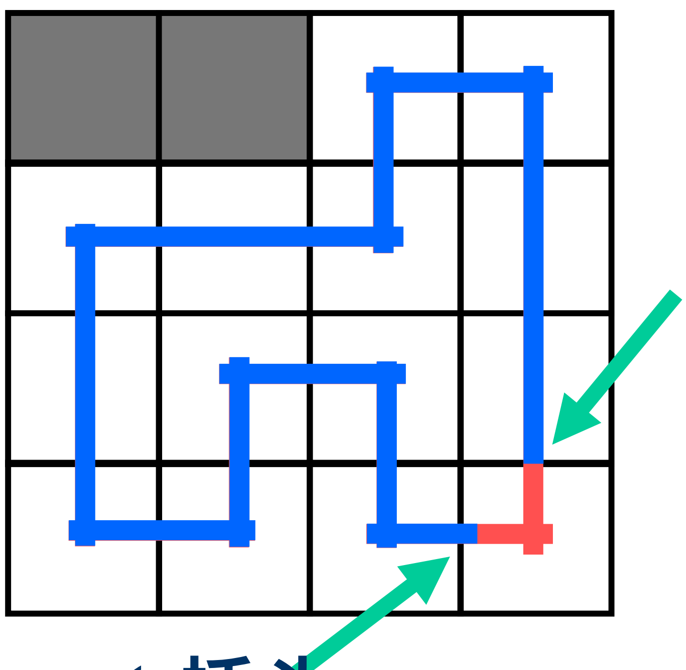
(# # #)

转移时间： $O(1)$

Case 2

有上插头和左插头

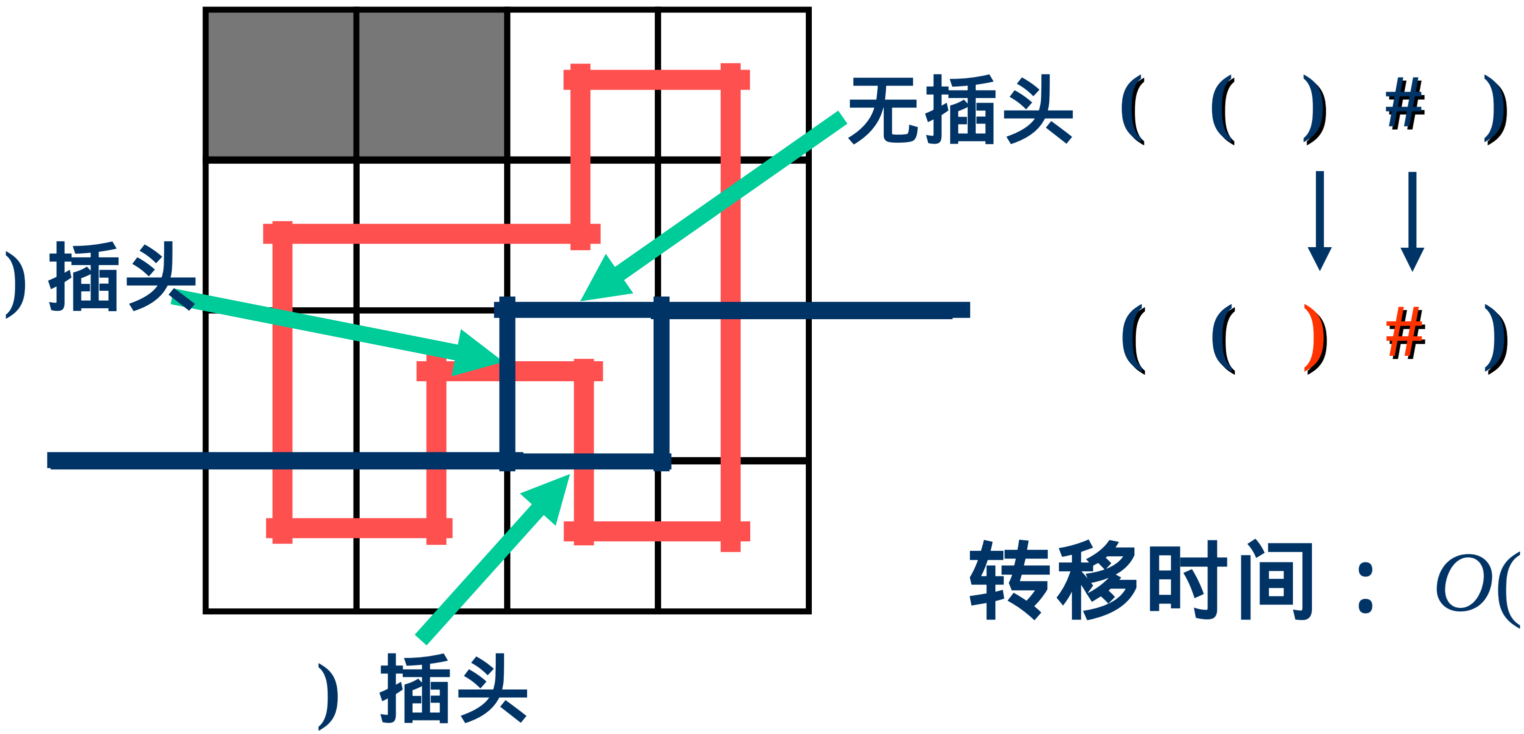
Case 2.3 左插头为 (插头，上插头为) 插头



路径的两端连接起来形成回路

Case 3

上插头和左插头恰好有一个，这种情况相当于延续原来的连通分量



转移时间： $O(1)$

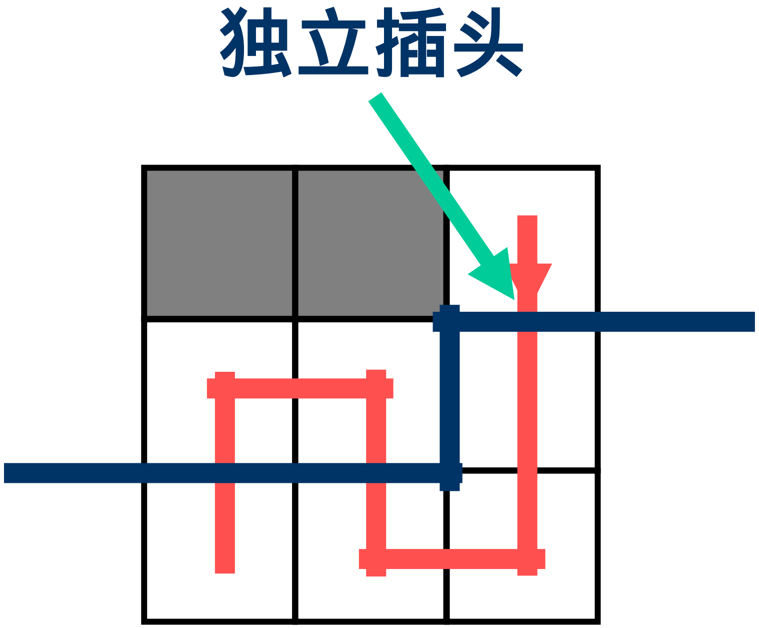
实验比较

测试数据	最小表示 7Based	最小表示 8Based	括号表示 3Based	括号表示 4Based
$m = n = 10$ 无障碍	31ms	15ms	0ms	0ms
$m = n = 11$ (1,1) 为障碍	187ms	109ms	46ms	31ms
$m = n = 12$ 无障碍	873ms	**499ms**	265ms	**140ms**

建议使用 2^k 进制，位运算效率高

拓展

- 如果求经过所有非障碍格子的**哈密顿路径**的个数呢？



3 进制→ 4 进制

- 0 → 无插头状态
- 1 → 左括号插头
- 2 → 右括号插头
- 3 → 独立插头

广义的括号匹配

- 如果一个连通块只有 1 个插头或大于 2 个插头呢？
- 括号表示法需要满足一个条件：每个连通块内恰好有 2 个插头.
- 对于一个大于 2 个插头的连通块

  最左边的插头标记为 (

  最右边的插头标记为)

  中间的插头标记为)(

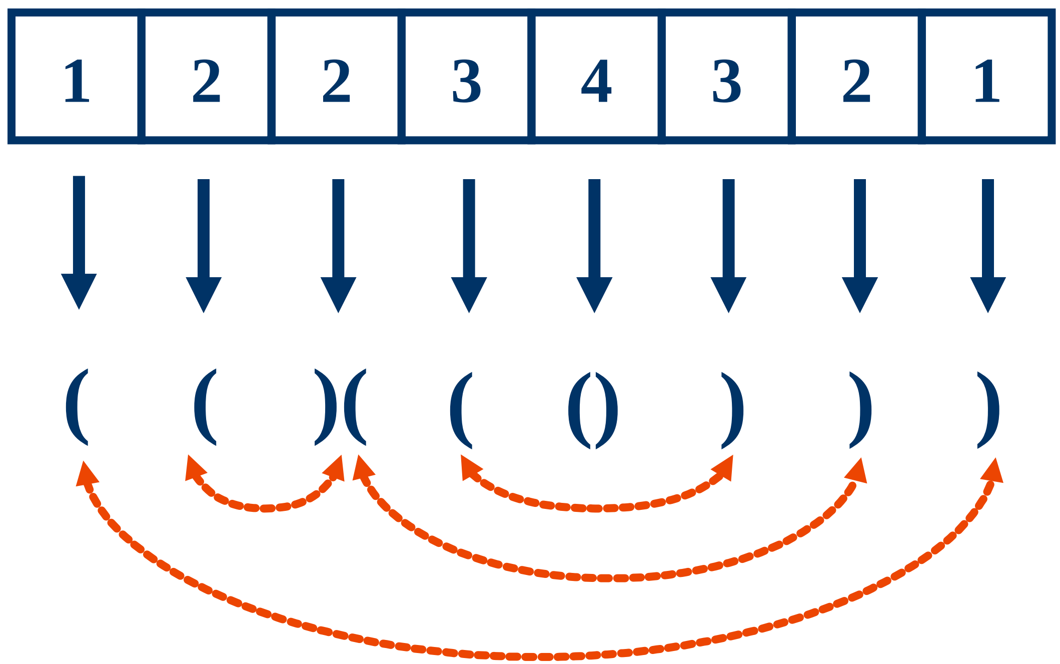
特殊性

- 单独为一个连通块的插头标记为 ( )

广义的括号表示法

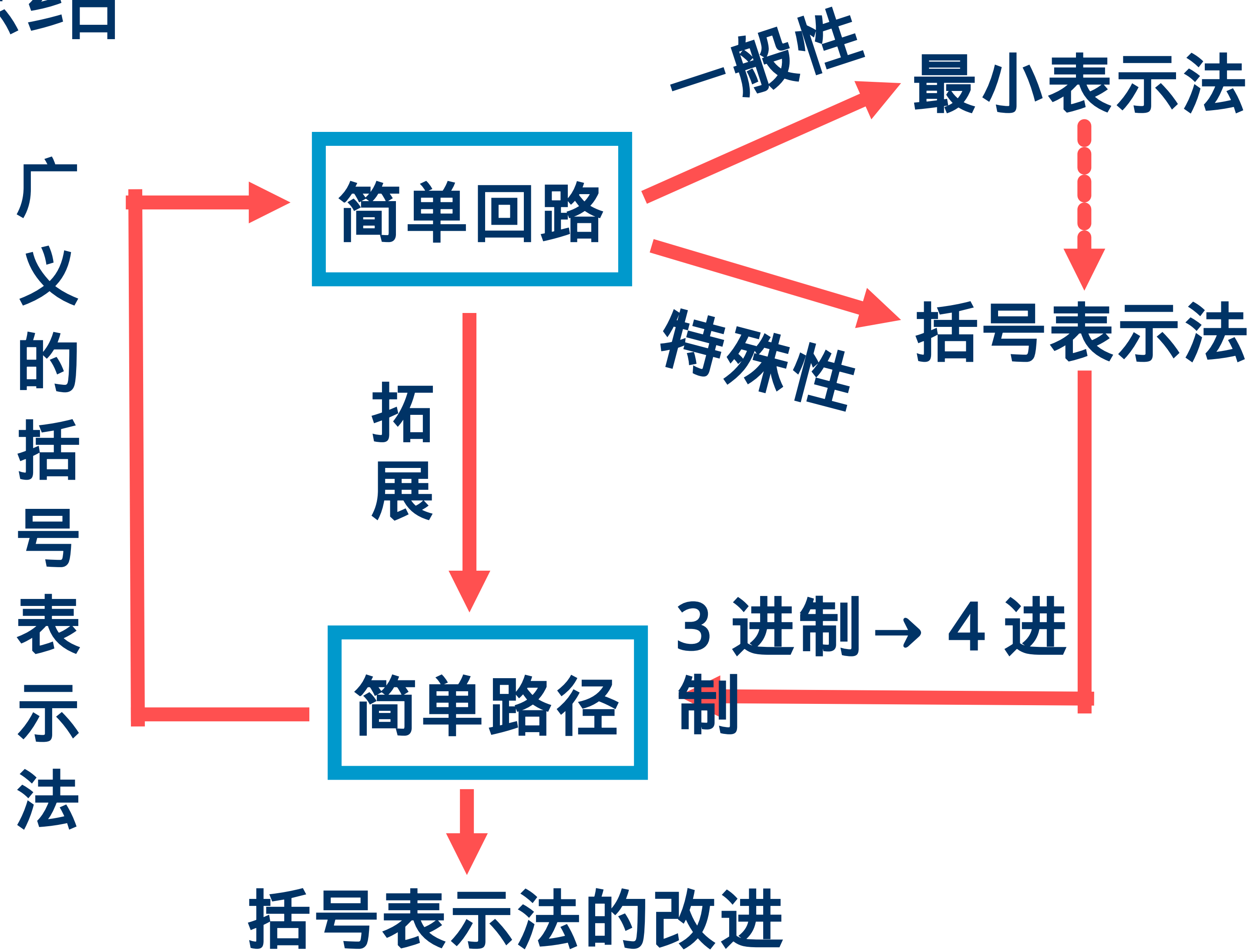
广义的括号表示法

- 左括号与右括号匹配对应的插头连通
- 例： 最小表示法 → 广义括号表示法



普适性

总结



全文研究内容

- **一类简单路径问题**

  Formula 1(Ural1519)

  Formula 2(改编自 Formula 1)

- **一类棋盘染色问题**

  Black & White(Uva10532)

- **一类基于非棋盘模型的问题**

  生成树计数 (NOI2007)

- **一类最优性问题的剪枝优化**

  Rocket Mania (Zju2125)

Thank you for listening!

Questions are welcome.