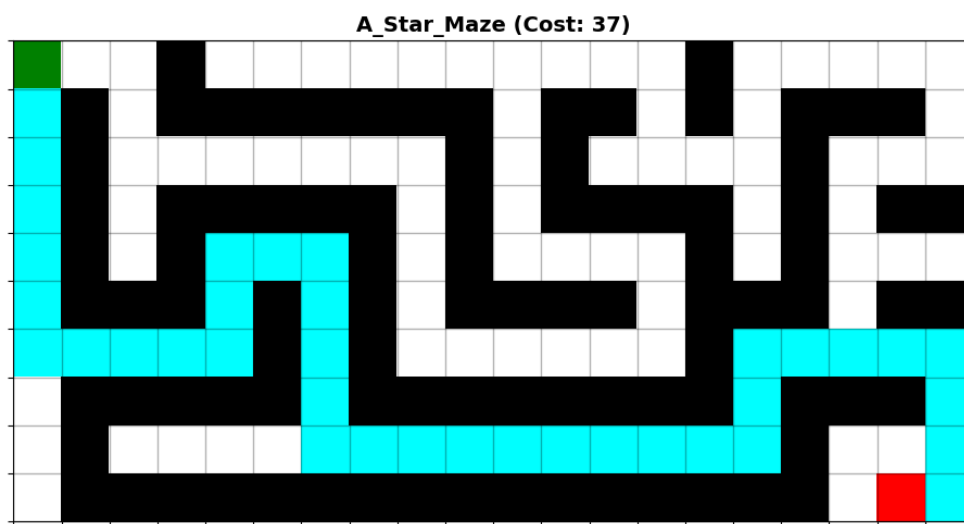


## 5.1 Path Cost / Length (ความเหมาะสมของเส้นทาง)

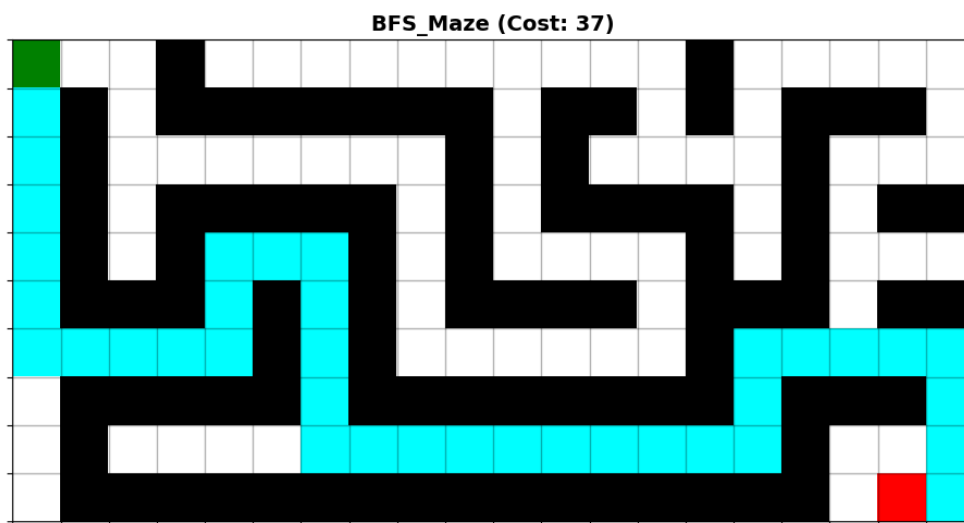
- BFS และ A Search:\* ทั้งสองอัลกอริทึมค้นพบเส้นทางที่ เหมาะสมที่สุด (Optimal) กล่าวคือ ได้ระยะทางที่สั้นที่สุดเท่าที่เป็นไปได้ (Shortest Path) เสมอ
- DFS: ค้นพบเส้นทางที่ ไม่เหมาะสมที่สุด (Not Optimal) เส้นทางที่ได้มักจะมีความยาวมาก มีลักษณะอ้อมไปมา หรือเดินจนสุดขอบแผนที่ ก่อนจะวกกลับมาหาเป้าหมาย เนื่องจากพฤติกรรมที่พยายามเจาะลึกลงไปเรื่อยๆ (Greedy approach)

## 5.2 Number of Nodes Expanded (ประสิทธิภาพการประมวลผล)

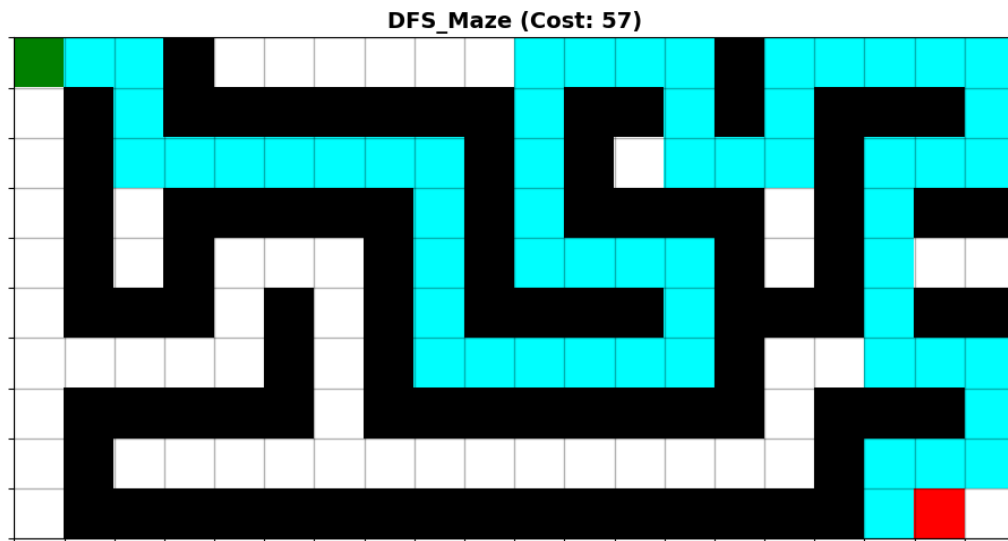
- A Search:\* มีประสิทธิภาพสูงสุด โดยมีการขยาย Node น้อยที่สุด เนื่องจากมีการใช้ค่า Heuristic ช่วยนำทาง ทำให้การค้นหามุ่งตรงเข้าหาเป้าหมาย ลดการสำรวจพื้นที่ที่ไม่จำเป็น



- BFS: มีประสิทธิภาพต่ำในการจัดการทรัพยากร โดยมีการขยาย Node จำนวนมาก เนื่องจากเป็นการค้นหาแบบตาบอด (Uninformed Search) ที่ต้องกระจายการค้นหาออกไปรอบทิศทางเป็นวงกว้างเหมือนหยดน้ำ



- **DFS:** จำนวน Node ที่ขยายไม่แน่นอน ขึ้นอยู่กับลำดับการเลือกทิศทาง แต่โดยรวมมักจะมีการสำรวจพื้นที่ที่ไม่เกี่ยวข้องจำนวนมากหากเลือกทางผิดตั้งแต่ต้น



### 5.3 วิเคราะห์ A Search และคุณภาพของ Heuristic\*

สาเหตุที่ A Search\* มีประสิทธิภาพดีกว่า BFS และ DFS ในสถานการณ์นี้ เนื่องจากหลักการทำงานของฟังก์ชัน  $f(n) = g(n) + h(n)$

1. **บทบาทของ Heuristic:** การนำค่า  $h(n)$  (ในที่นี้คือ Manhattan Distance) มาใช้ เปรียบเสมือนการมี "เข็มทิศ" ที่คอยชี้ทิศทางไปยังเป้าหมาย ทำให้ A\* ไม่ต้องสุ่มเดินสะเปะสะปะเหมือน BFS และไม่หลงทางลึกเหมือน DFS
2. **ความถูกต้อง (Admissibility):** Manhattan Distance เป็น Heuristic ที่ดี (Admissible) เพราะค่าที่คำนวณได้ ไม่เคยสูงเกินระยะทางจริง ทำให้ A\* มั่นใจได้ว่าจะเจอเส้นทางที่สั้นที่สุดเสมอ
3. สถานการณ์ที่ A อาจแย่กว่า:\*
  - หากเลือกใช้ Heuristic ที่ไม่ดี (Overestimation): เช่น คำนวณค่า  $h(n)$  สูงเกินความจริง A\* อาจจะมองข้ามเส้นทางที่ดีที่สุดไป ทำให้ได้คำตอบที่ไม่ Optimal
  - ความซับซ้อนของการคำนวณ: หากสูตรคำนวณ Heuristic ซับซ้อนและใช้เวลาประมวลผลนานมาก อาจทำให้เวลาโดยรวมช้ากว่าการค้นหาแบบง่ายๆ อย่าง BFS ได้