

迷宫问题

1. 算法运行结果

1.1 bfs

Total cost: 89.

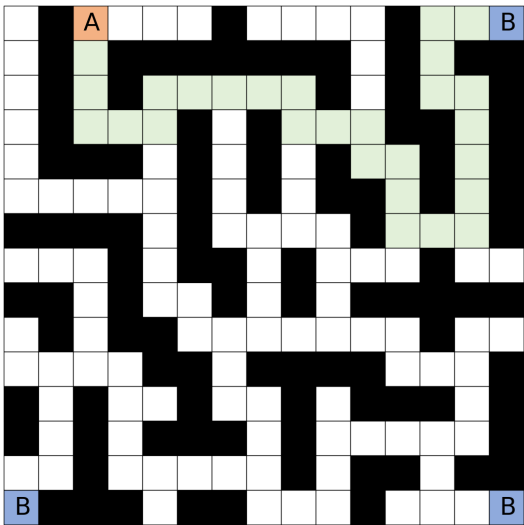


Figure 1

1.2 dfs

Total cost: 120.

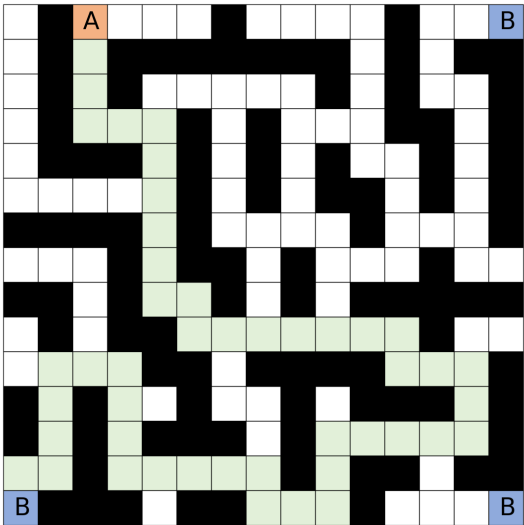


Figure 2

1.3 iddfs

Total cost: 82.

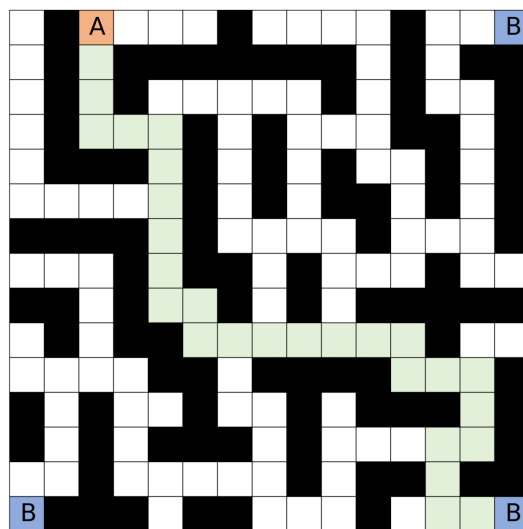


Figure 3

1.4 ucs

Total cost: 69.

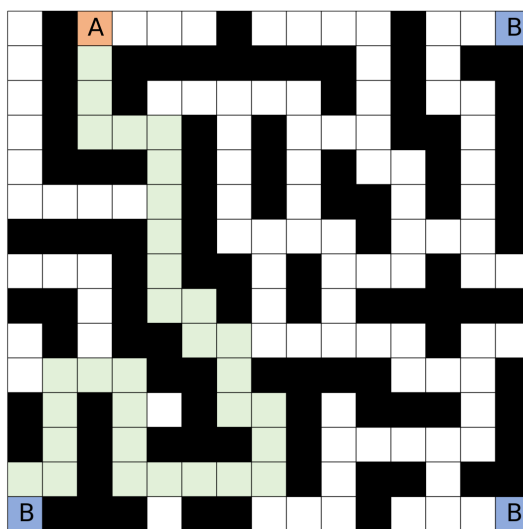


Figure 4

1.5 astar

Total cost: 69.

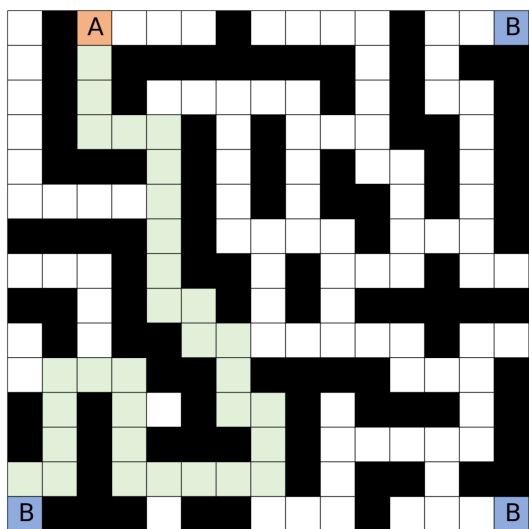


Figure 5

2. 结果的比较和说明

2.1 结果比较

1. **BFS**: 总代价89。BFS按层扩展，确保最短步数，但在带权图中无法保证最小代价。
2. **DFS**: 总代价120。DFS优先深入搜索，容易陷入高代价路径，不保证最优性。
3. **IDDFS**: 总代价82。通过逐步增加深度限制，IDDFS在内存受限时接近最优解，但代价仍高于UCS和A*。
4. **UCS**: 总代价69。UCS优先扩展最小累积代价的节点，确保最小代价路径，代价最优。
5. **A***: 总代价69。结合启发式函数，A*在保证最优性的同时减少搜索范围，效率高于UCS。

2.2 算法性质

- **BFS**: 最优性（最短步数），时间复杂度 $O(b^d)$ ，空间复杂度 $O(b^d)$ ，适用于无权图。
- **DFS**: 非最优，时间复杂度 $O(b^m)$ ，空间复杂度 $O(m)$ ，适合内存受限场景。
- **IDDFS**: 最优性，时间复杂度 $O(b^d)$ ，空间复杂度 $O(d)$ ，平衡内存与最优性。
- **UCS**: 最优性（最小代价），时间复杂度 $O(b^{(1 + C/\epsilon)})$ ，空间复杂度 $O(b^d)$ ，适用于带权图。
- **A***: 最优性（可采纳启发式），时间复杂度 $O(b^d)$ （优化），空间复杂度 $O(b^d)$ ，适合有高质量启发式的场景。

3. 启发式函数

其核心逻辑如下：

1. **初始化**: 使用优先队列 `PriorityQueueFrontier` 管理前沿节点，起点 `init_node` 的A*成本 `f(n)` 初始化为启发式函数 `h(n)` 的值。

2. 节点扩展：在 `process_new_node` 中，计算新节点 `new_node` 的A*成本 $f(n) = g(n) + h(n)$ ，其中 `g(n)` 是从起点到当前节点的实际成本，`h(n)` 是启发式估计值。

3. 前沿管理：

- 如果新节点不在前沿中，将其加入优先队列，并记录父节点、路径成本和A*成本。
- 如果新节点已在前沿中且新成本更低，则更新其优先级和父节点。

4. 启发式函数：通过 `heuristic_func` 提供启发式估计值，确保算法高效性和最优性。

4. 改变上下左右移动代价

变成1, 1, 1, 1后：

bfs: 28

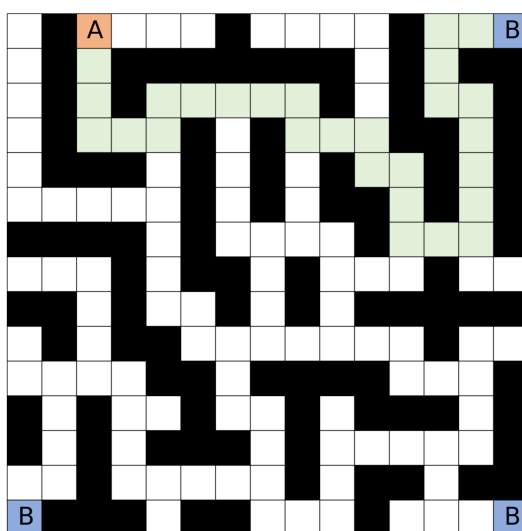


Figure 6

dfs: 46

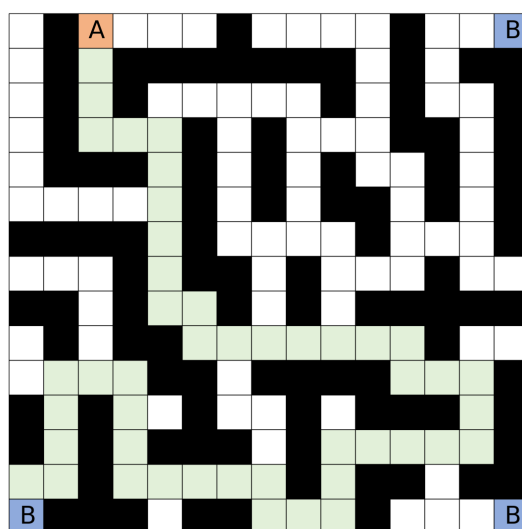


Figure 7

iddfs: 28

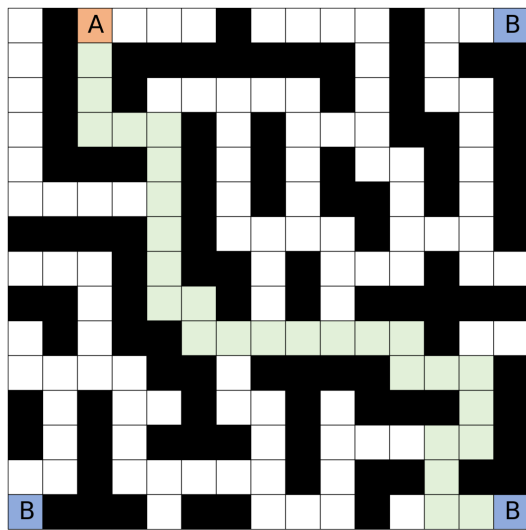


Figure 8

ucs: 28

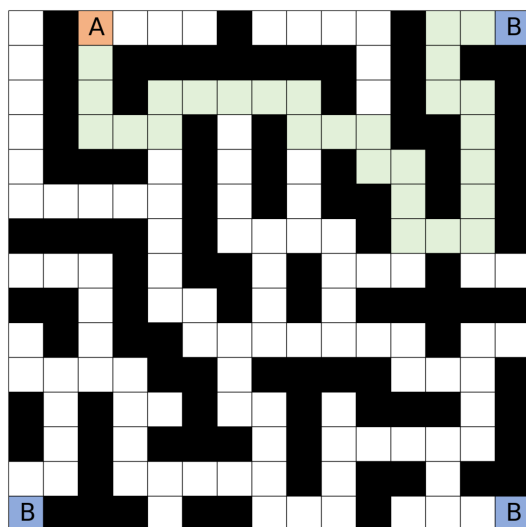


Figure 9

astar: 28

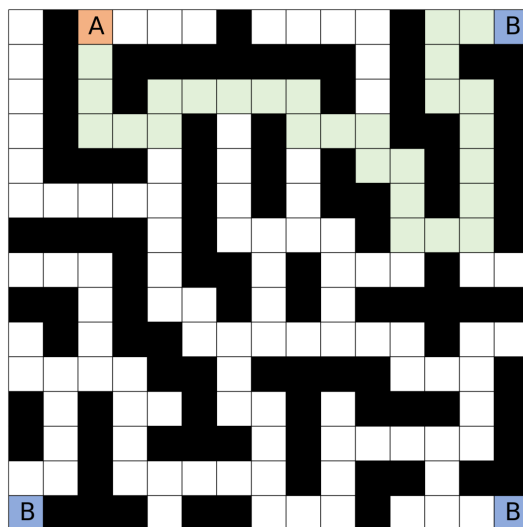


Figure 10

当上下左右移动代价均为1时，BFS、IDDFS、UCS、A*) 均输出最优路径，总代价为28，表明在无权图中这些算法均能保证最短路径。DFS因无最优性，输出较高代价（46）。此时，UCS和A*退化为BFS，IDDFS通过深度限制也达到最优。