Exercise 17

1. Depth-first search always expands at least as many nodes as A\* search with an admissible heuristic.

**错**。DFS可能沿最短路径直接找到目标结点，仅需扩展**d**（目标结点深度）个结点；而带有可采纳的启发式的A\*算法可能因启发式引导选择探索周边结点以保证找到最优解。所以DFS扩展的结点数可能少于A\*算法。

1. h(n)=0 is an admissible heuristic for the 8-puzzle.

**对**。因为代价总是**非负**的，所以结点n的实际代价一定大于等于h(n)=0，所以h(n)一定是可采纳的。是最弱的启发式，相当于无启发。

1. A is of no use in robotics because percepts, states, and actions are continuous.

**错**。虽然机器人感知和状态空间通常是连续的，但可通过离散化操作将问题转化为A\*适用的离散状态空间，所以A\*算法在机器人学中仍然有重要作用。

1. Breadth-first search is complete even if zero step costs are allowed.

**对**。BFS的完备性与其遍历策略和状态空间的性质相关，与边成本无关。若存在解，即使允许零成本边，BFS仍能保证在有限分支因子的状态空间中找到解。

（完备性：若有解，一定能找到解）

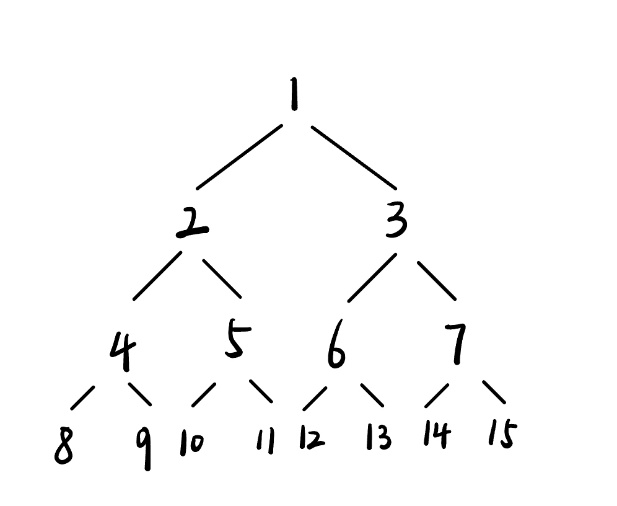
1. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

**错**。此问题中的代价是移动次数，移动一次最多可以消除一个方向上的任意差距，而曼哈顿距离刻画的是两跟方格的距离大小，二者无直接联系，甚至曼哈顿距离往往会过高估计代价。例如，在以A、B为对角点的矩形边上无棋子阻挡时，曼哈顿距离一定大于等于2，而实际代价始终为2。综上，曼哈顿距离在此问题中是不可采纳的。

Exercise 18

Consider a state space where the start state is number 1 and each state kk has two successors: numbers 2k and 2k+1.  
1. Draw the portion of the state space for states 1 to 15.  
2. Suppose the goal state is 11. List the order in which nodes will be visited for breadth-first search, depth-limited search with limit 3, and iterative deepening search.  
3. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?  
4. Does the answer to (c) suggest a reformulation（重形式化） of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?  
5. Call the action going from kk to 2k Left, and the action going to 2k+1 Right. Can you find an algorithm that outputs the solution to this problem without any search at all?

1. 状态空间如下：



1. 各算法访问结点顺序如下：

BFS: 1-2-3-4-5-6-7-8-9-10-11

DLS(limit 3): 1-2-4-8-9-5-10-11

IDS: 1; 1-2-3; 1-2-4-5-3-6-7; 1-2-4-8-9-5-10-11

1. 双向搜索在此问题中效率较高，因为状态空间为严格二叉树，结构简单，反向路径唯一，可以避免冗余搜索。

正向分支因子为2，反向分支因子为1。

1. 是。可以从目标结点开始，由于状态空间为严格二叉树，只能进行单向回溯，所以可以无需搜索，直接沿唯一路径到达结点1。
2. 将状态进行二进制编码，结点1编码为1，通过向编码右侧添加数来代替分支选择操作。规定添加0代表选择向左（选择2k分支），添加1代表向右（选择2k+1分支），如结点2的编码为10，结点5的编码为101。

那么对任意目标结点n，其二进制编码的数值即等于n的二进制表示，也就可以直接通过二进制转换得到从1到达目标结点的路径

Exercise 25

Prove each of the following statements, or give a counterexample:  
1. Breadth-first search is a special case of uniform-cost search.  
2. Depth-first search is a special case of best-first tree search.  
3. Uniform-cost search is a special case of A search.

1. **对。**当UCS中所有边的代价相等时，到每个结点的成本即等于该结点的深度，此时UCS的优先级队列会像BFS按层扩展的行为一致。
2. **对。**Best-first tree搜索按照评估函数f(n)选择代价最小的结点进行扩展，若令

**f(n)=-depth(n)**，那么Best-first tree搜索会优先扩展深度最大的结点，此时即与DFS的行为一致。

1. **对。**A\*搜索的评估函数为**f(n)=g(n)+h(n)**。而当 **h(n)=0 时**，**f(n)=g(n)**，此时A搜索完全按照路径成本扩展节点，与UCS的行为一致。

Exercise 26

Compare the performance of A and RBFS on a set of randomly generated problems in the 8-puzzle (with Manhattan distance) and TSP (with MST—see ) domains. Discuss your results. What happens to the performance of RBFS when a small random number is added to the heuristic values in the 8-puzzle domain?

|  |  |  |
| --- | --- | --- |
| **8-puzzle (with Manhattan distance)** | | |
| 算法名称 | **优势** | **缺点** |
| **A\*算法** | **·**使用可采纳且一致的启发式保证了最优解。  **·**优先队列系统性地扩展最低成本路径，扩展节点数较少，效率高。 | 内存占用高（需存储所有已生成节点的优先队列） |
| **RBFS** | 内存占用少（仅存储当前路径及兄弟节点的回溯信息）。 | **·**节点扩展数通常多于A\*（因回溯导致重复扩展）。  **·**递归调用带来的额外开销不可忽略。 |

|  |  |  |
| --- | --- | --- |
| **TSP (with MST—see )** | | |
| 算法名称 | **优势** | **缺点** |
| **A\*算法** | 使用MST启发式可高效估计剩余路径成本，引导搜索方向，从而系统性地找到最优解，减少扩展节点数。 | 内存占用高（需存储所有已生成节点的优先队列） |
| **RBFS** | **·**内存占用少  **·**可通过递归剪枝避免存储全部节点。 | **·**启发式计算频繁（每次回溯需重新评估兄弟节点），会增加计算时间。  **·**对启发式误差敏感，可能导致更多回溯。 |

综合比较之下，在小规模问题中A\*算法的效率显著优于RBFS，但在大规模问题或内存受限的情况下，RBFS可能因为内存占用小的优势而更可行。

**在8-puzzle启发式中添加小随机数对RBFS的影响：**

若该随机扰动破坏了启发式的可采纳性，可能导致RBFS扩展无关节点，造成扩展的结点数增加，甚至可能导致RBFS无法找到最优解而找到次优解。

Exercise 36

Invent a heuristic function for the 8-puzzle that sometimes overestimates, and show how it can lead to a suboptimal solution on a particular problem. (You can use a computer to help if you want.) Prove that if h never overestimates by more than c, A\* using h returns a solution whose cost exceeds that of the optimal solution by no more than c.

**h(n) = Manhattan(n)+misplace(n)**，其中misplace(n)表示状态n下错位的数字个数，显然这个启发式h有时会高估代价。

证明：设C\*表示最优解实际代价，设h(n)≤h\*(n)+c，h\*是状态n到目标的实际代价。

设G2为一个次优解，其实际代价比最优解G1的实际代价多c，即 g(G2)>C\*+c。

现考虑到达最优解的路径上的任意结点n，有

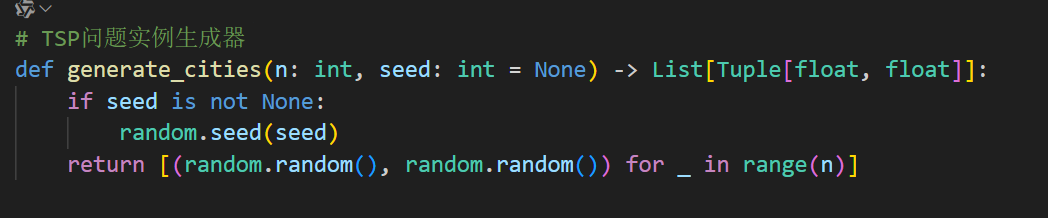
**f(n)= g(n)+ h(n)** ≤ **g(n)+ h\*(n)+c** ≤ **C\*+c** < **g(G2)**

故在G1总是先于G2被扩展，那么任何实际代价比C\*(G1)多c的次优解都不可能被扩展，结论得证。

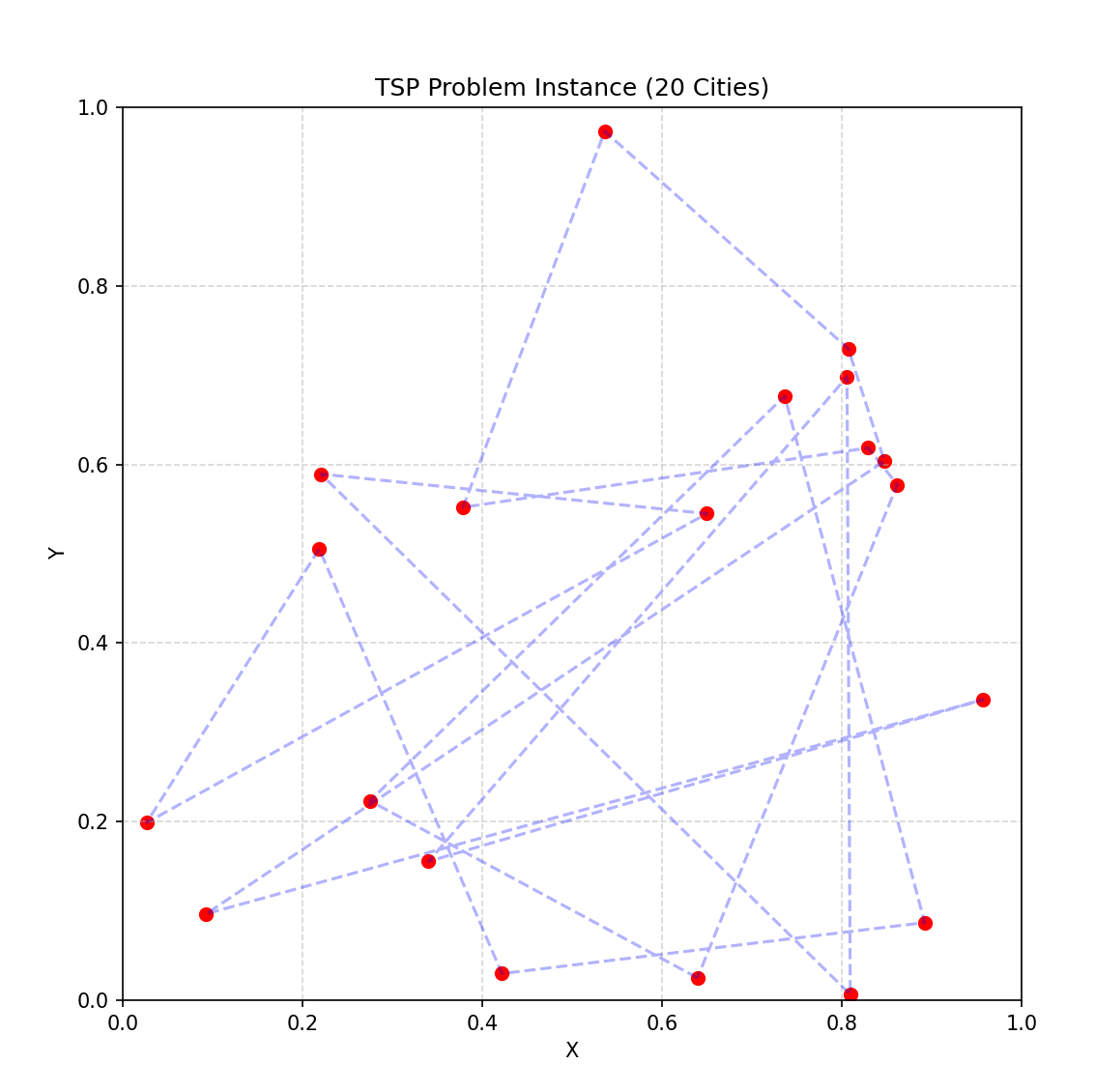
Exercise 38

The traveling salesperson problem (TSP) can be solved with the minimum-spanning-tree (MST) heuristic, which estimates the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.  
1. Show how this heuristic can be derived from a relaxed version of the TSP.  
2. Show that the MST heuristic dominates straight-line distance.  
3. Write a problem generator for instances of the TSP where cities are represented by random points in the unit square.  
4. Find an efficient algorithm in the literature for constructing the MST, and use it with A graph search to solve instances of the TSP.

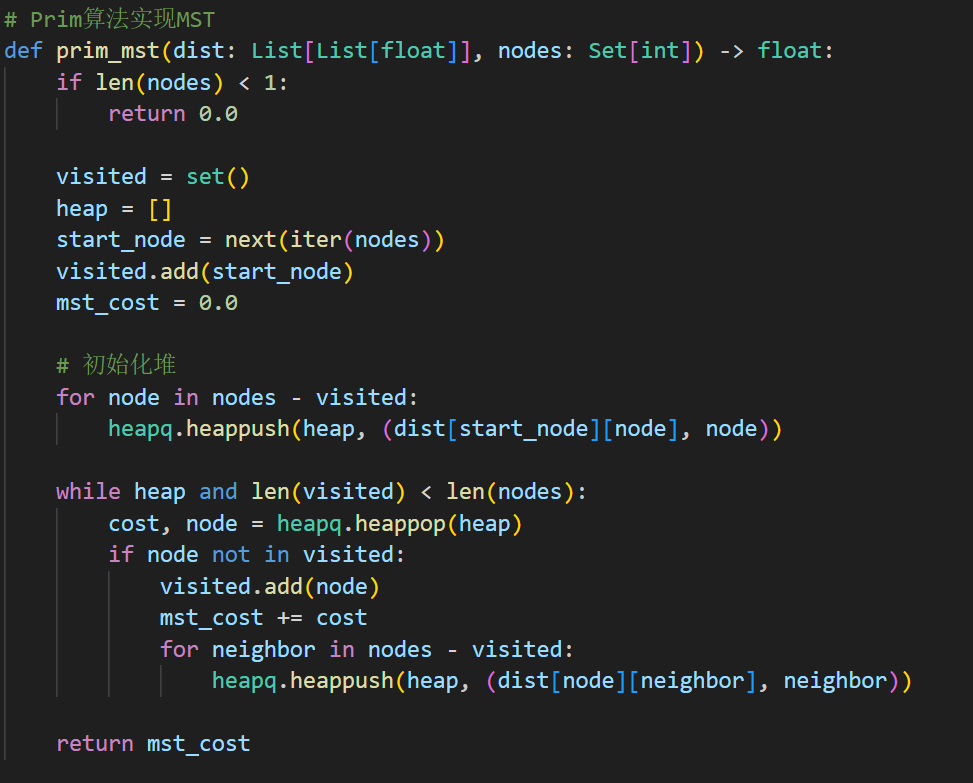
1. TSP问题是找到一条穿过城市的最小路径，形成一个闭环。MST在此问题中会生成一个连接所有城市的无环图​（即树），它的总长度总是小于或等于闭环。故MST是可接受的。
2. MST给出的从当前结点到目标结点的距离必然是**一条直线距离**或**多条直线距离的和**，在城市之间的距离满足**三角不等式**的情况下，MST给出的距离必然大于两座城市的直线距离。
3. **TSP问题实例生成器**

**源码截图**：

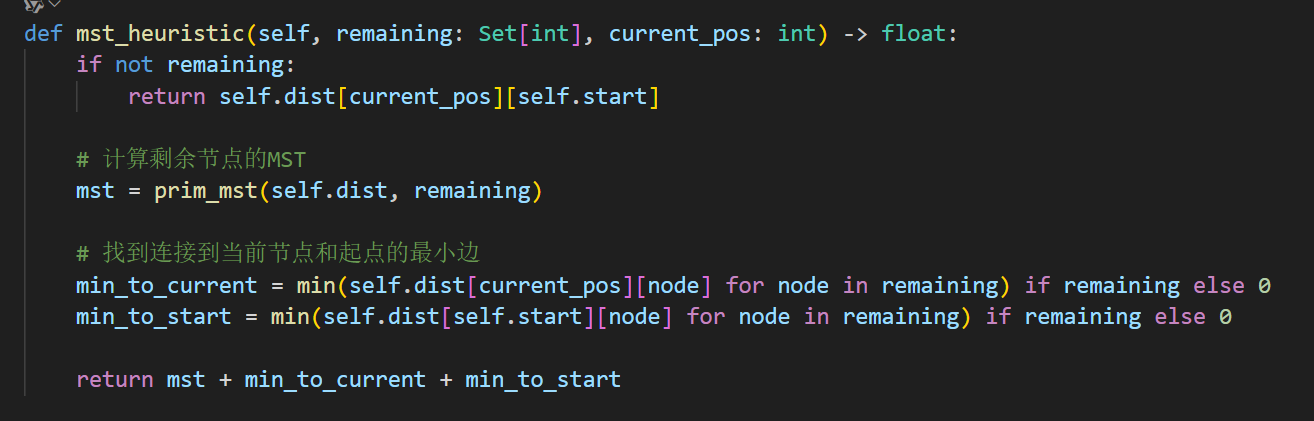
**可视化运行结果**：

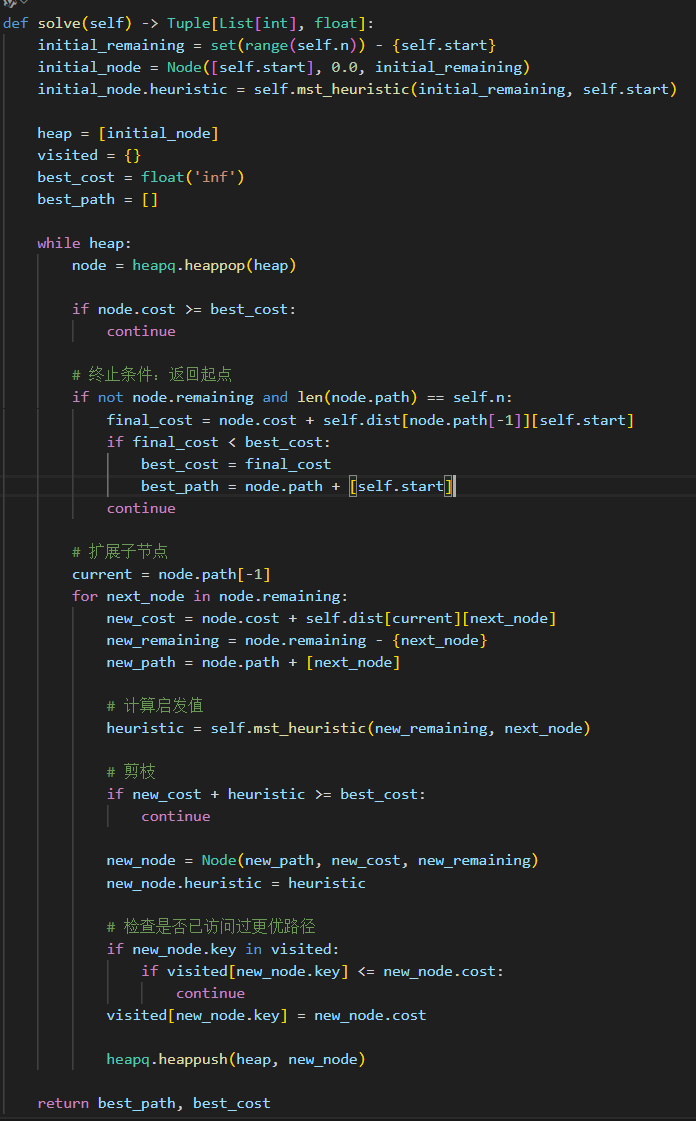


1. **MST的构造使用了prim算法，源码如图：**

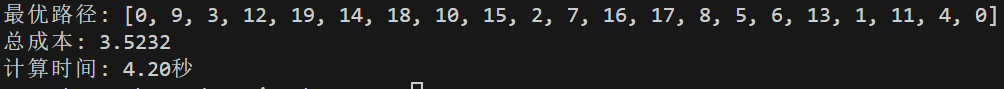


**A\*搜索的启发式函数及求解器：**

****



**运行结果如下：**

****