人工智能实验报告

姓名：陈宇轩

学号：ZF2121109

实验题目：Python实现A\*算法

实验目的：使用Python实现A\*算法，加深对算法的理解

实验内容：使用Python实现A\*算法的核心思想，并用matplotlib包画出A\*搜索过程。

核心代码：

def searching(self):

"""

A\_star Searching.

:return: path, visited order

"""

self.PARENT[self.s\_start] = self.s\_start

self.g[self.s\_start] = 0

self.g[self.s\_goal] = math.inf

heapq.heappush(self.OPEN,(self.f\_value(self.s\_start), self.s\_start))

while self.OPEN:

\_, s = heapq.heappop(self.OPEN)

self.CLOSED.append(s)

if s == self.s\_goal: # stop condition

break

for s\_n in self.get\_neighbor(s): # exploring new point

new\_cost = self.g[s] + self.cost(s, s\_n)

if s\_n not in self.g:

self.g[s\_n] = math.inf

if new\_cost < self.g[s\_n]: # conditions for updating Cost

self.g[s\_n] = new\_cost

self.PARENT[s\_n] = s

heapq.heappush(self.OPEN, (self.f\_value(s\_n), s\_n))

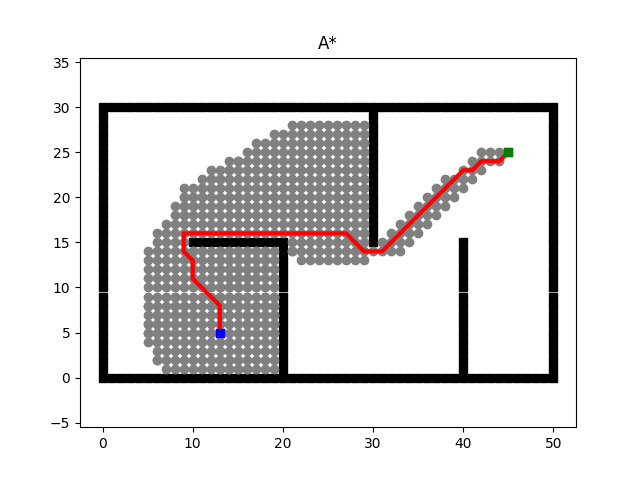
return self.extract\_path(self.PARENT), self.CLOSED

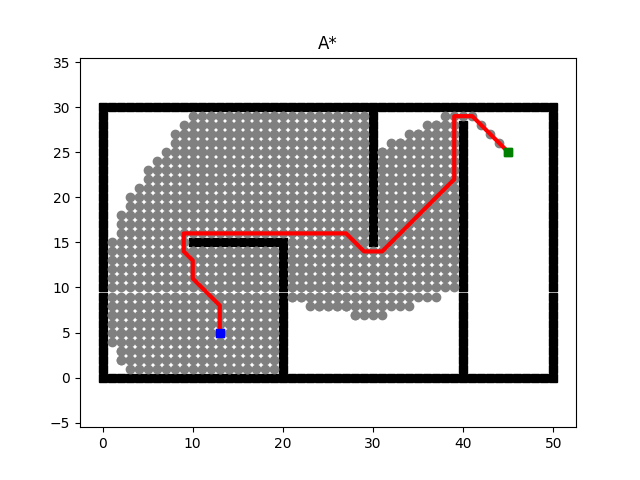
其中使用heapq（大顶堆或优先队列）保存每个遍历的路径点，get\_neighbor()用于发现相邻路径点（没有被障碍物阻挡），cost()函数用于计算路径代价f，f= g+h 其中g为到达某一处的代价，h为该点的启发式代价，h的计算规则既可以使用曼哈顿距离公式，也可以使用欧几里得距离公式。

在2D平面下，算法的基本思想是从起点开始，计算到达与起点相邻的八个没有障碍物的位置，并不断向四周扩散，过程中，优先考虑f代价最小的路径。

结束的条件是找到目标点，或者搜索完空间中可行的路径点。

实验结果截图：





实验收获：

与传统的深度优先或者其他非启发式算法相比，A\*算法在寻找目标路径的时候，用更高效的方式（对路径点进行了代价评估）对搜索路径进行了优化，从而提升了算法的性能。

但当出现极端情况时，比如终点不可达，A\*算法却不能及时停止，会按部就班的计算整个空间，造成一定程度上计算资源的浪费。