作业4.2 – 1133730117 - 张实唯

## 实验结果

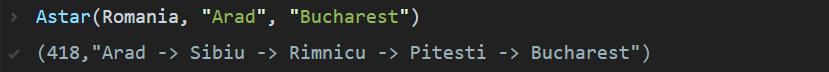
1. **哈工大数据集**

实际最佳路线为 "正心楼 -> 3G体验店 -> 工商银行 -> 诚意楼"



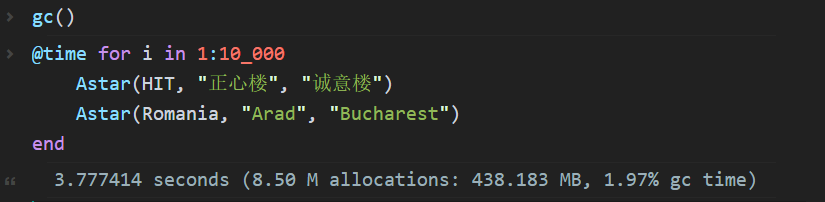
1. **罗马尼亚数据集**

实际最佳路线为 "Arad -> Sibiu -> Rimnicu -> Pitesti -> Bucharest"



1. **运行速度**

使用A\*算法运行上述两个测试各10000次，记录总时间。



1. **小结**

BFS、UCS、DFS、IDS的测试结果见上次作业(4.1)报告。

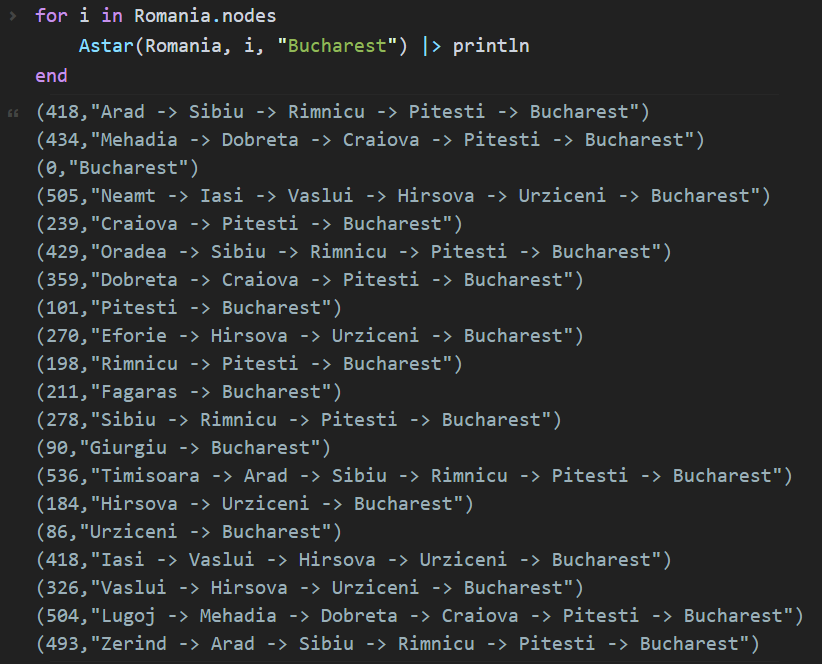
|  |  |  |  |
| --- | --- | --- | --- |
| 算法 | 完备性\* | 最优性\*\* | 时空开销\*\*\* |
| BFS | 是 | 否 | 3.4 |
| UCS | 是 | 是 | 4.6 |
| DFS | 是 | 否 | 3.1 |
| IDS | 是 | 否 | 4.1 |
| Astar | 是 | 是 | 3.8 |

\*两个测试中均能找到正确路径

\*\*两个测试中均能找到最优路径

\*\*\*运行两个测试10000次所需时间(秒)，由于语言使用GC管理内存，难以测定空间开销

1. **不同起点到Bucharest的最短路径**



## 代码解释

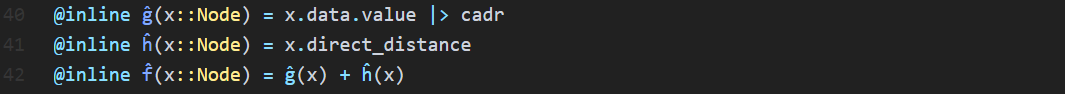
1. **核心算法**

Astar和UCS代码只有3行区别，即最内层的if语句中对代价和优先级的计算方式增加了每个节点的估计代价ĥ，我将UCS中的对应部分用注释的形式附在了代码中，可以直观对比。



1. **代价计算**

使用数据集中存储的每个节点到Bucharest的直线距离(Node.direct\_distance)作为估计值。



1. **路径生成**

和BFS及UCS相似，我使用每个节点的data指针来追踪扩展出这个节点的上一个节点，因此在算法结束后从目标节点出发顺着data指针就可以输出路径。

