**Assignment 1 Part 2 Search**

**Zhichao He z5282955**

Question 1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Start10 | Start12 | Start20 | Start30 | Start40 |
| UCS | 2565 | Mem | Mem | Mem | Mem |
| IDS | 2407 | 13812 | 5297410 | Time | Time |
| A\* | 33 | 26 | 915 | Mem | Mem |
| IDA\* | 29 | 21 | 952 | 17297 | 112571 |

Analysis:

*b: average branching factor d: depth of the shallowest solution*

UCS Dijkstra is based on BFS, thus the time complexity and space complexity are both . So, it’s easy for it to run out of memory as step of moves becomes large.

IDS’s time complexity is , but the space complexity is . Thus, it will not run out of memory very soon. But time will still grow up rapidly as step of moves become large.

A\* search avoid a lot of meaningless branch of states so the nodes expanded will decrease a lot. But it stores all the nodes expanded in memory, so space complexity is , and is possible to run out of memory.

IDA\* improves A\* by using DFS iteratively and stores only nodes, so it’s unlikely to run out of memory unless step of moves is very large.

Question 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Start50 | | Start60 | | Start64 | |
| IDA\* | 50 | 14642512 | 60 | 321252368 | 64 | 1209086782 |
| 1.2  1.4  1.6 | 52  66  100 | 191438  116342  33504 | 62  82  148 | 230861  4432  55626 | 66  94  162 | 431033  190278  235848 |
| Greedy | 164 | 5447 | 166 | 1617 | 184 | 2174 |

(b) I modified the part of calculating f(n) function in the code.

The original code was

F1 is G1 + H1,

If w = 1.2, which means

Thus, I changed the code to

G1\_ is G1 \* 0.8,

H1\_ is H1 \* 1.2,

F1 is G1\_ + H1\_,

To change the implementation of .

(d) Analysis:

IDA\* uses , and can always find the best solution. But the time cost is very high when step of moves is very large.

Greedy search uses . In this scenario, it can find the solution much faster no matter how complex the problem is. But the solution may not be optimal.

The new versions of IDA\* use , where . If , it’s IDA\* and if , it’s Greedy search. These 3 algorithms combine the advantages of IDA\* and Greedy search. They have some optimality and save some time.

Thus, if we must find the optimal solution then IDA\* is preferred. If we focus on finding the solution very fast, then Greedy search is preferred. If we both want to save time and have some optimality, then we can change the value and find a best to meet our need.