CS 4320/5314

Programming Assignment 1:

Search and Pathfinding

The following were our findings and results for the implemented search algorithms. The following results were made using randomly generated maps with increasing sizes.

**Breadth First Search:**

**5x5**

Cost: 15

Number of nodes expanded: 21

Max number of nodes held in memory: 5

Runtime in milliseconds: 0.1220703125

Path sequence: (4, 3) [3 3] [2 3] [2 2] [1 2] [0 2]

**10x10**

Cost: 33

Number of nodes expanded: 89

Max number of nodes held in memory: 9

Runtime in milliseconds: 0.4019737243652344

Path sequence: (6, 9) [6 8] [6 7] [6 6] [6 5] [6 4] [6 3] [6 2] [6 1] [6 0] [5 0] [4 0] [3 0] [2 0]

A picture containing calendar

Description automatically generated

**15x15**

Cost: 15

Number of nodes expanded: 183

Max number of nodes held in memory: 16

Runtime in milliseconds: 0.5350112915039062

Path sequence: (4, 8) [5 8] [5 7] [5 6] [5 5] [5 4] [5 3]

**20x20**

Cost: 52

Number of nodes expanded: 322

Max number of nodes held in memory: 18

Runtime in milliseconds: 0.9369850158691406

Path sequence: (13, 10) [12 10] [11 10] [10 10] [10 9] [9 9] [8 9] [7 9] [6 9] [6 8] [6 7] [6 6] [6 5] [5 5] [4 5] [3 5] [2 5] [2 4] [2 3]

**Iterative Deepening Search:**

**5x5**

Cost: 19

Number of nodes expanded: 27

Runtime in milliseconds: 0.1049041748046875

Path sequence: (4, 1) [3, 1] [2, 1] [1, 1] [0, 1] [0, 2]

Chart, scatter chart

Description automatically generated

**10x10**

Cost: 22

Number of nodes expanded: 114

Runtime in milliseconds: 2.379179000854492

Path sequence: (3, 5) [4, 5] [4, 6] [4, 7] [4, 8] [3, 8] [2, 8] [1, 8]

**15x15**

Cost: 35

Number of nodes expanded: 119

Runtime in milliseconds: 155.63273429870605

Path sequence: (9, 4) [9, 5] [8, 5] [7, 5] [7, 6] [7, 7] [7, 8] [7, 9] [6, 9] [5, 9] [4, 9]

**20x20**

Cost: 30

Number of nodes expanded: 189

Runtime in milliseconds: 50.733089447021484

Path sequence: (19, 7) [19, 8] [19, 9] [19, 10] [18, 10] [18, 11] [18, 12] [18, 13] [17, 13] [16, 13] [15, 13]

**A\* Search:**

**5x5**

Cost: 18

Number of nodes expanded: 14

Max number of nodes held in memory: 10

Runtime in milliseconds: 0.2770423889160156

Path sequence: (1, 4) [1 3] [1 2] [1 1] [2 1] [3 1] [4 1]

**10x10**

Cost: 18

Number of nodes expanded: 117

Max number of nodes held in memory: 19

Runtime in milliseconds: 1.2328624725341797

Path sequence: (8, 5) [7 5] [7 4] [6 4] [5 4] [5 3] [4 3] [4 2] [4 1] [3 1]

**15x15**

Cost: 29

Number of nodes expanded: 118

Max number of nodes held in memory: 24

Runtime in milliseconds: 1.1739730834960938

Path sequence: (3, 14) [ 2 14] [ 1 14] [ 1 13] [ 1 12] [ 1 11] [ 1 10] [1 9] [2 9] [2 8] [2 7] [3 7]

A picture containing background pattern

Description automatically generated

**20x20**

Cost: 100

Number of nodes expanded: 297

Max number of nodes held in memory: 64

Runtime in milliseconds: 3.002166748046875

Path sequence: (3, 17) [ 3 16] [ 3 15] [ 2 15] [ 2 14] [ 1 14] [ 0 14] [ 0 13] [ 0 12] [ 0 11] [ 0 10] [0 9] [0 8] [0 7] [0 6] [0 5] [0 4] [1 4] [1 3] [2 3] [2 2] [3 2] [3 1] [4 1] [5 1] [6 1] [7 1] [8 1] [9 1] [9 2] [10 2] [11 2] [11 3] [12 3] [13 3] [14 3] [14 2] [14 1] [15 1]

**Results**

Given the results mentioned above, we came into the following conclusions:

For the cost of the path, BFS had a great increase in cost when the size of the map increased. IDS and A\* had interestingly low costs even as the size increased. There was an outlier for the 20x20 map size cost in A\*, as it greatly increased from its previous iterations.

There was a trend for number of nodes expanded, as they all seemed to increase in size as the map grew bigger. With no clear differences between each pathfinding algorithm.

The team had difficulty collecting the maximum number of nodes held in memory for the IDS algorithm due to its recursive nature. However, BFS had lower numbers than A\*, which kept increasing along map size, while BFS numbers barely increased.

There was a clear weak algorithm in terms of time, which was IDS. The other algorithms had incredibly fast speeds, but the clear winner was BFS, in which all iterations were less than 1 millisecond.