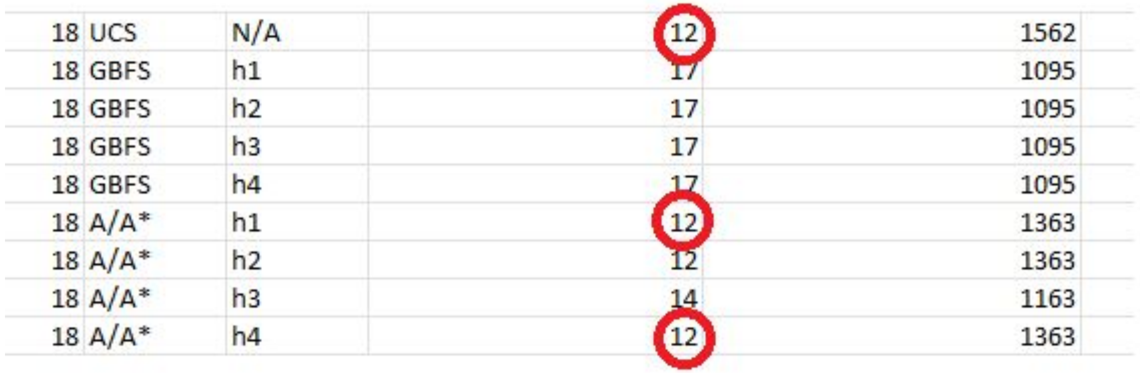
# Comp 472 Mini Project 2 Analysis

By Lisa Duong (40097705) & Deokyeong Kim (26942105)

## 1. The length of the solutions across algorithms and heuristics. When do you have the lowest-cost solution?

- Observing our results, GBFS has lengths of solution paths that are greater or equal to the other search algorithms. This may be because GBFS will find a solution quickly, but it does not guarantee the lowest cost solution path.

- According to our results, we have the lowest-cost solution for Uniform Cost Search and Algorithm A with h1 and h4. Uniform Cost Search is sorted by its cost and looks for the lowest cost. Algorithm A with h1 and h4 is Algorithm A*(see further slides); hence, it also guarantees the lowest cost solution path.

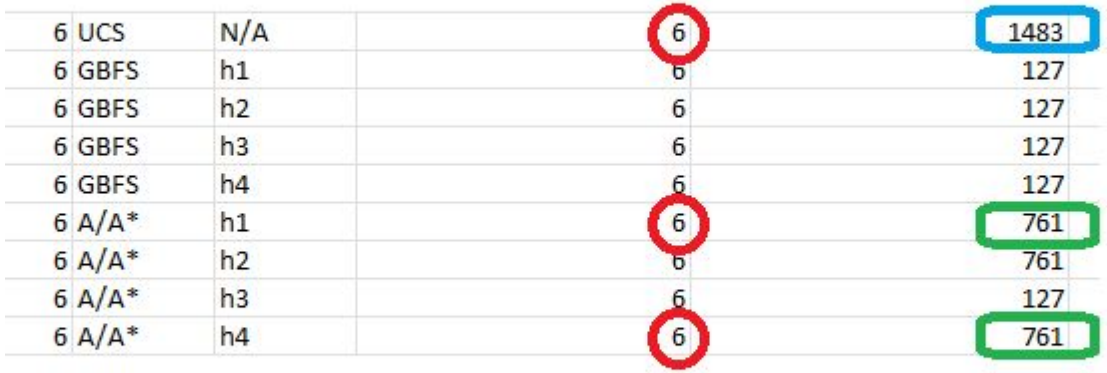


| | | | | |
|---|---|---|---|---|
| 18 | UCS | N/A | 12 | 1562 |
| 18 | GBFS | h1 | 17 | 1095 |
| 18 | GBFS | h2 | 17 | 1095 |
| 18 | GBFS | h3 | 17 | 1095 |
| 18 | GBFS | h4 | 17 | 1095 |
| 18 | A/A* | h1 | 12 | 1363 |
| 18 | A/A* | h2 | 12 | 1363 |
| 18 | A/A* | h3 | 14 | 1163 |
| 18 | A/A* | h4 | 12 | 1363 |

## 2. The admissibility of each heuristic and its influence on the optimally of the solution.

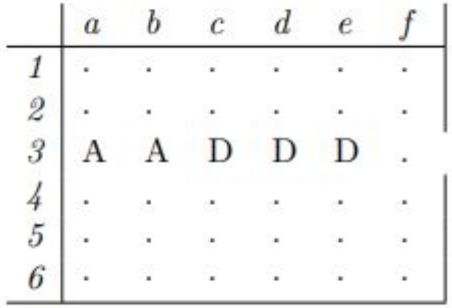
- **h1:** h1 is admissible because blocking vehicles need to move at least once which means the actual cost is at least one cost per blocking vehicles and another cost for the ambulance to reach the exit. The actual cost would always be greater than the heuristic. $h(n) < h^*(n)$

- Its influence: Algorithm A using h1 becomes Algorithm A* because it is admissible and the g(n) (cost) can be greater or equal to g*(n) (lowest cost of start to node). Hence, it guarantees the lowest cost solution path but it has a smaller search path than Uniform Cost Search.



| | | | | |
|---|---|---|---|---|
| 6 | UCS | N/A | 6 | 1483 |
| 6 | GBFS | h1 | 6 | 127 |
| 6 | GBFS | h2 | 6 | 127 |
| 6 | GBFS | h3 | 6 | 127 |
| 6 | GBFS | h4 | 6 | 127 |
| 6 | A/A* | h1 | 6 | 761 |
| 6 | A/A* | h2 | 6 | 761 |
| 6 | A/A* | h3 | 6 | 127 |
| 6 | A/A* | h4 | 6 | 761 |

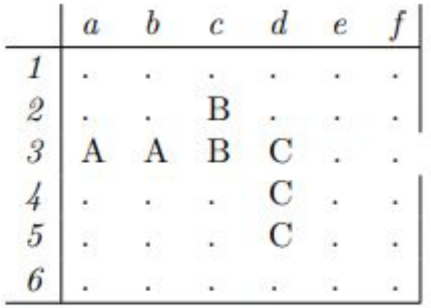
- **h2**: h2 is not admissible. Imagine the case where one car is blocking ambulance horizontally, which has length three. Then h2 is equal to the length of that vehicle. However the actual cost to goal is only 2 since ambulance can exit right after the vehicle exits.



|   | *a* | *b* | *c* | *d* | *e* | *f* |
|---|---|---|---|---|---|---|
| *1* | . | . | . | . | . | . |
| *2* | . | . | . | . | . | . |
| *3* | A | A | D | D | D | . |
| *4* | . | . | . | . | . | . |
| *5* | . | . | . | . | . | . |
| *6* | . | . | . | . | . | . |

- Its influence: h2’s influence on algorithms does not differ much from algorithms with h1 as the only difference is in the execution time of GBFS where it only differs slightly. Because of their similar nature, algorithms ran with h1 and h2 share similar results in many cases.

- **<u>h3:</u>** h3 is not admissible. Imagine the case of h1, where two cars are blocking the ambulance. In this case, h1 is 2, and the actual cost to the goal is 3. Whichever lambda (>1) is chosen to multiply h1, the h3 will be greater than the actual goal.



|   | *a* | *b* | *c* | *d* | *e* | *f* |
|---|---|---|---|---|---|---|
| *1* | . | . | . | . | . | . |
| *2* | . | . | B | . | . | . |
| *3* | A | A | B | C | . | . |
| *4* | . | . | . | C | . | . |
| *5* | . | . | . | C | . | . |
| *6* | . | . | . | . | . | . |

- <u>Its influence</u>: h3's influence is only seen in algorithm A, as it is sorted by f(n) where f(n) is g(n) + h(n). Here in h3, the lambda puts much more emphasis on the heuristic. The max h1 in our project can be is 4 as there can only be max 4 vehicles that can block the ambulance whereas the cost can continue to increase as we visit more nodes. Therefore, the cost will have more impact when we use h1. However, in h3 with the lambda, we can get a higher and more drastic heuristic which makes it the deciding factor.

## Example for h3’s influence

Algorithm A with h1:

g(n) : 1 + h(n) : 3 = f(n) : 4 <- We would pick this one since it is lower

g(n) : 3 + h(n) : 2 = f(n) : 5

Algorithm A with h3 where λ = 5:

g(n) : 1 + h(n) : λ*3 = f(n) : 16

g(n) : 3 + h(n) : λ*2 = f(n) : 13 <- We would pick this one since it is lower

## h4: Our chosen admissible heuristic

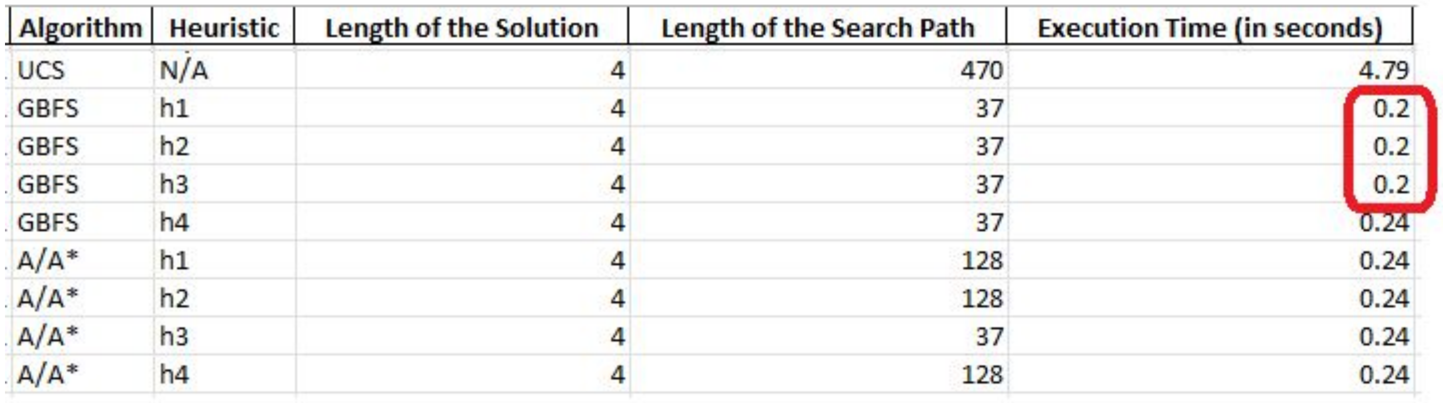
- **h4:** The h4 that we decided to go with is **h1 + 1**. It is admissible for similar reasons as h1. The blocking vehicles would need to move at least once and the ambulance would have to use 1 cost to reach the exit. Thus, h4 will always be smaller or equal to the actual cost for the ambulance to reach the exit.

- Its influence: It's is same reason as h1.

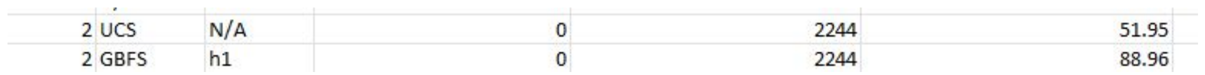
## 3. The execution time across algorithms and heuristics. Is an informed search always faster?

- From our results, we can see that in most of the puzzles, the execution times of GBFS is the fastest among the three search algorithms. It is based on heuristic which is the estimate of a node to the goal. We can also see that the lengths of the search paths of puzzles with GBFS have the smallest numbers which indicates that it has visited fewer states than the others resulting in a shorter execution time.

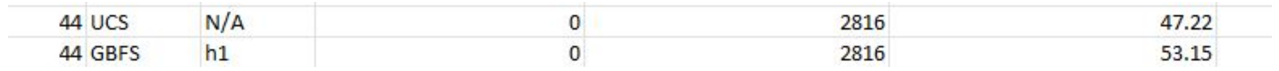
- Informed search is usually faster; however, it is not always faster in the cases where we have puzzles with no solution because Uniform Cost Search (uninformed search) seems to have the shortest execution time. This may be because the Uniform Cost Search that we implemented in this project is a Breadth First Search (BFS) (explained in further slides). It exhausts all successors in each layer systematically compared to the other algorithms. Since the cost is always the same in each layer as the cost is always increasing by one, there would not be a case where we would have to replace a state in the open list. Whereas in other algorithms, each successor generated may be possibly replaced in open list.



| Algorithm | Heuristic | Length of the Solution | Length of the Search Path | Execution Time (in seconds) |
|---|---|---|---|---|
| UCS | N/A | 4 | 470 | 4.79 |
| GBFS | h1 | 4 | 37 | 0.2 |
| GBFS | h2 | 4 | 37 | 0.2 |
| GBFS | h3 | 4 | 37 | 0.2 |
| GBFS | h4 | 4 | 37 | 0.24 |
| A/A* | h1 | 4 | 128 | 0.24 |
| A/A* | h2 | 4 | 128 | 0.24 |
| A/A* | h3 | 4 | 37 | 0.24 |
| A/A* | h4 | 4 | 128 | 0.24 |



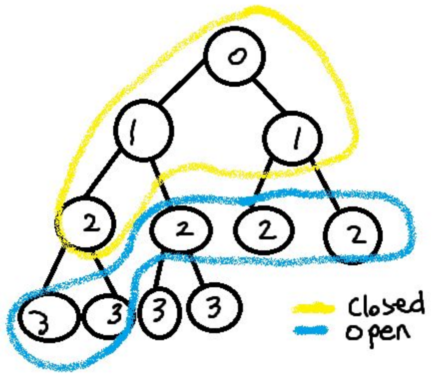
| | | | | | |
|---|---|---|---|---|---|
| 2 | UCS | N/A | 0 | 2244 | 51.95 |
| 2 | GBFS | h1 | 0 | 2244 | 88.96 |



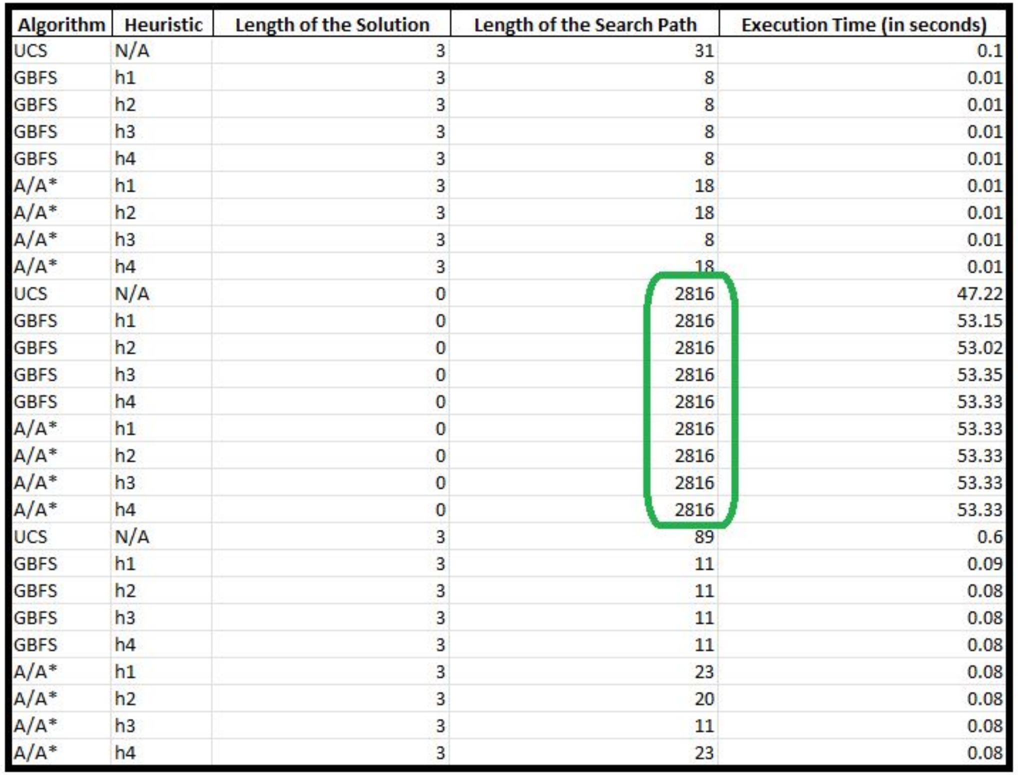
| | | | | | |
|---|---|---|---|---|---|
| 44 | UCS | N/A | 0 | 2816 | 47.22 |
| 44 | GBFS | h1 | 0 | 2816 | 53.15 |

## 4. Other interesting facts that you deem worthy of describing.

- For our uniform cost search the cost increase by one each time it generates successors and all successors generated from same parent node have same cost. Thus, UCS visits all its siblings before visiting next successors. Thus, our UCS is acting like a breadth-first search (BFS) because all the same level nodes are visited before the next level nodes.



- No solution cases tend to have much larger search path length compare to regular boards. Because every state should be found and make sure there's no solution. All algorithm exhausted all the possible states as shown in the image below.



| Algorithm | Heuristic | Length of the Solution | Length of the Search Path | Execution Time (in seconds) |
|---|---|---|---|---|
| UCS | N/A | 3 | 31 | 0.1 |
| GBFS | h1 | 3 | 8 | 0.01 |
| GBFS | h2 | 3 | 8 | 0.01 |
| GBFS | h3 | 3 | 8 | 0.01 |
| GBFS | h4 | 3 | 8 | 0.01 |
| A/A* | h1 | 3 | 18 | 0.01 |
| A/A* | h2 | 3 | 18 | 0.01 |
| A/A* | h3 | 3 | 8 | 0.01 |
| A/A* | h4 | 3 | 18 | 0.01 |
| UCS | N/A | 0 | 2816 | 47.22 |
| GBFS | h1 | 0 | 2816 | 53.15 |
| GBFS | h2 | 0 | 2816 | 53.02 |
| GBFS | h3 | 0 | 2816 | 53.35 |
| GBFS | h4 | 0 | 2816 | 53.33 |
| A/A* | h1 | 0 | 2816 | 53.33 |
| A/A* | h2 | 0 | 2816 | 53.33 |
| A/A* | h3 | 0 | 2816 | 53.33 |
| A/A* | h4 | 0 | 2816 | 53.33 |
| UCS | N/A | 3 | 89 | 0.6 |
| GBFS | h1 | 3 | 11 | 0.09 |
| GBFS | h2 | 3 | 11 | 0.08 |
| GBFS | h3 | 3 | 11 | 0.08 |
| GBFS | h4 | 3 | 11 | 0.08 |
| A/A* | h1 | 3 | 23 | 0.08 |
| A/A* | h2 | 3 | 20 | 0.08 |
| A/A* | h3 | 3 | 11 | 0.08 |
| A/A* | h4 | 3 | 23 | 0.08 |