Brandon Pauly

CSC 380

Assignment 1

Youtube: http://youtu.be/VjaD8tcpsKw

Outputs:

EASY	Length	Time	Memory
DFS	14 moves	0 millis	11 nodes
BFS	6 moves	16 millis	18 nodes
Greedy	60 moves	594 millis	34275 nodes
A*	6 moves	0 millis	12 nodes

MEDIUM	Length	Time	Memory
DFS	730 moves	31 millis	638 nodes
BFS	10 moves	15 millis	109 nodes
Greedy	10 moves	16 millis	194 nodes
A*	10 moves	0 millis	112 nodes

HARD	Length	Time	Memory
DFS	21189 moves	187 millis	18763 nodes
BFS	31 moves	1094 millis	24971 nodes
Greedy	31 moves	16 millis	51 nodes
A*	31 moves	1314 millis	24969 nodes

Analysis:

As we can see from the output on easy, Greedy Best performs the worst. In this case it is largely just a matter of circumstance I believe. A* and BFS are very similar, although A* performs the best of all. DFS is not bad either, it produces a solution quickly here and uses the least amount of memory, but produces a slightly worse solution in terms of optimality.

On medium we see output that is more along the lines of what we would expect. DFS is the worst in general, giving us a very long solution after twice as much time as the other algorithms and uses markedly more space. Other than that we see what must be the optimal solution for the other three, with space and time complexities that are very much aligned in terms of magnitude.

On hard we see some anomalous data again. Here in DFS, we see a quick solution in terms of time complexity, and the length of the solution is about what we would expect to see and the memory usage is fair for how many steps we are looking at. The other 3 algorithms all return the optimal solution in this case, but A* is starting to align a bit with BFS in this scenario. Greedy Best surprises, largely by getting lucky in its guesses, with a very lightweight traversal, blowing the other three algorithms out of the water.

In the end, Greedy Best's weakness can also be its benefit. It is chaotic because it is making the least educated guesses. Very unreliable in general. BFS and DFS are a bit of a trade-off. DFS produces the worst solution in most cases, and this is to be expected. It can, however, often beat out BFS in time. BFS, on the other hand, is optimal here; consistently producing the solution with the least number of steps. A* is the star though it seems. Most reliable overall. The educated guesses makes it the most stable of the four here.

Opinion Piece:

Do I think my algorithms are intelligent? No. I believe for the most part these algorithms are rather unintelligent. Many of them are very utilitarian in nature. When we got to A*, I started thinking we were heading down the right path, however. I can see how A* can be applied to different scenarios that possibly *would* yield an algorithm that I think is intelligent.

Furthermore, I will say that I think the problem is just too easy to be considered intelligent. Not that solving these problems and implementing this puzzle was easy, but the problem itself is just too simple to warrant such a consideration. I do not think that a machine requires a consciousness to be considered intelligent, but do I think that anything that gives me the impression that it is deciding on something could be. This is what I am starting to see in A*. While we are still doing this utility by looking at possibilities from each node, we are giving the algorithm a logical way to make a consideration. This is starting to mimic what people might do in such a scenario. "If I move this piece here, then I can move that one there, then that one goes there and I will be closer to a solution…" This is how we might look at the problem and break it down also. Although we probably come up with a solution a lot less of the time, that is because we cannot keep all of those states in memory. Sometimes neither can a computer, but they are a lot better at it.

This begs the question then: What makes something intelligent? Intelligence is sort of an ambiguous term that connotes one or several different things. I believe that intelligence is subjective and relative. Some might describe a dog as intelligent merely because it understands a lot of commands and is obedient, while others would say that is simply learned behavior through association. So while I think our little old rinky dink 8 puzzle solver isn't

something I would describe as intelligent, it lays the foundation for *real* intelligent algorithms. Algorithms that use several paradigms to make educated guesses at where the answer to the problem is most likely found. At the end of the day, that is exactly what we do. We don't always know where the answer will be, and sometimes we miss it, but we learn that our logic and intuition is typically the best place to start.