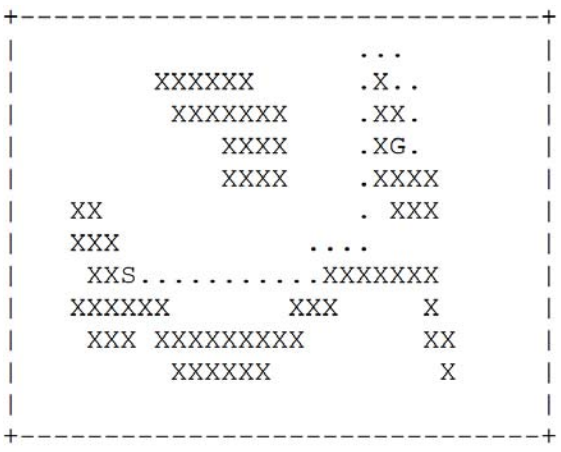
Sacha Raman
1288048

# COMPARISON OF A* AND BEST-FIRST

I have implemented the Best-First and A* algorithms and the following are the results I have obtained from running these algorithms on the three provided maps and on a map I have created.

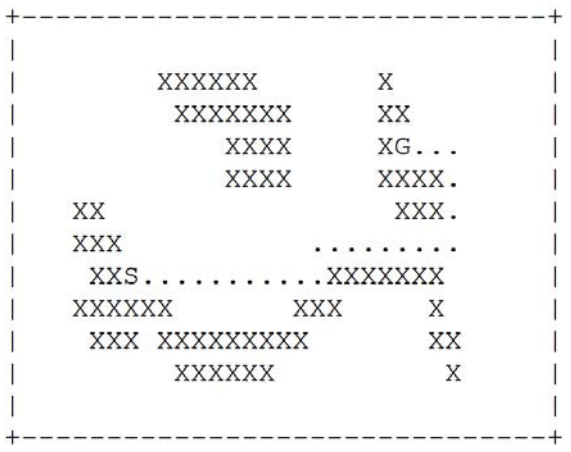
## MAP ONE

Best First



```
Number of state expansions: 66
Cost: 28
"Elapsed time: 41.14106 msecs"
```

A*

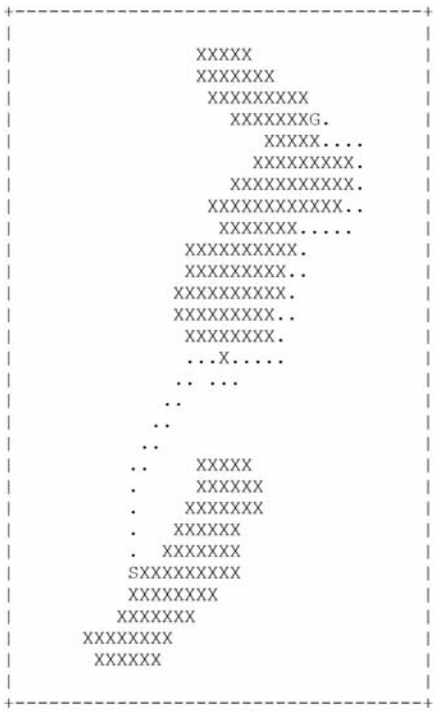


```
Number of state expansions: 408
Cost: 26
"Elapsed time: 411.400229 msecs"
```

A* found a marginally better path than Best-First did with the path length for A* being 26 from Start to Goal and 28 for Best-First. The route taken is the same until the end where Best-First choses to go up around the land containing Goal from the top whereas A* goes from below. A* takes around 10 times as long as Best-First to find a path as it computes a lot more expansions: 408 compared to the 66 expansions made by Best-First.

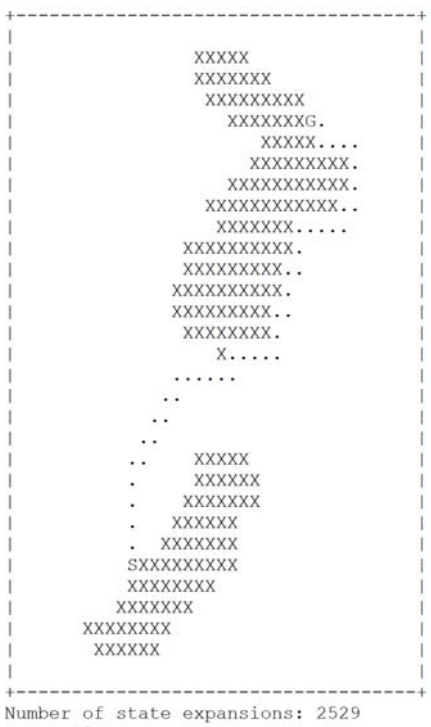
## MAP TWO

Best-First



```
Number of state expansions: 46
Cost: 47
"Elapsed time: 51.818087 msecs"
```

A*

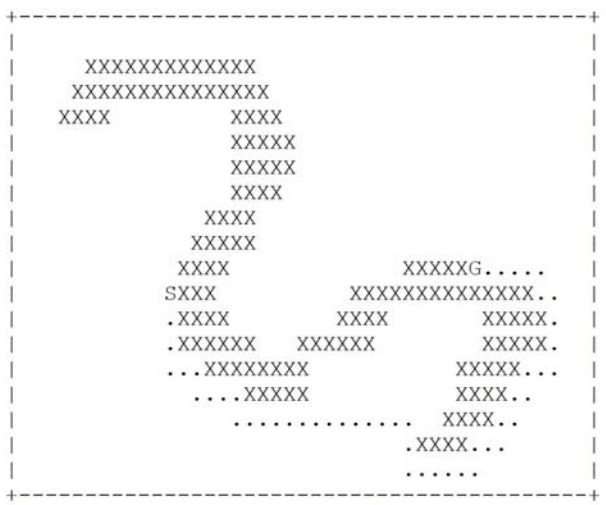


```
Number of state expansions: 2529
Cost: 45
"Elapsed time: 12654.237922 msecs"
```

Similar to Map One, the path returned by Best-First is not too different from the path returned by A*. The length of the path returned by Best-First was 47 whereas the length of the path returned by A* was 45. The only differences in the two paths occurs around the bottom of North Island where Best-First goes all the way up to the land and then follows the contours of the land as it is only considering the heuristic which is the Euclidean distances to Goal. On the other hand, A* sees going right up to the land's edge as a redundant move so it only goes as close up to the land as necessary and then starts veering right towards Goal. This is because A* takes the cost of the path into consideration as well as the heuristic value. Best-First only computes 46 expansions meanwhile A* makes 2529 expansions resulting in A* to take considerably longer than Best-First (51msec vs. 12654msec).

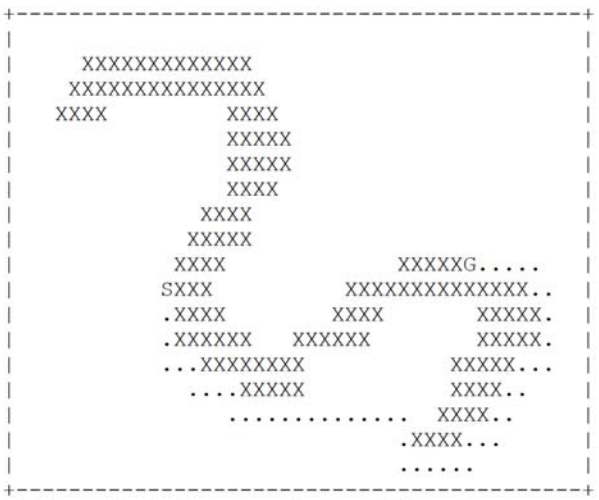
## MAP THREE

Best-First



```
Number of state expansions: 40721
Cost: 50
"Elapsed time: 18498.850608 msecs"
```

A*

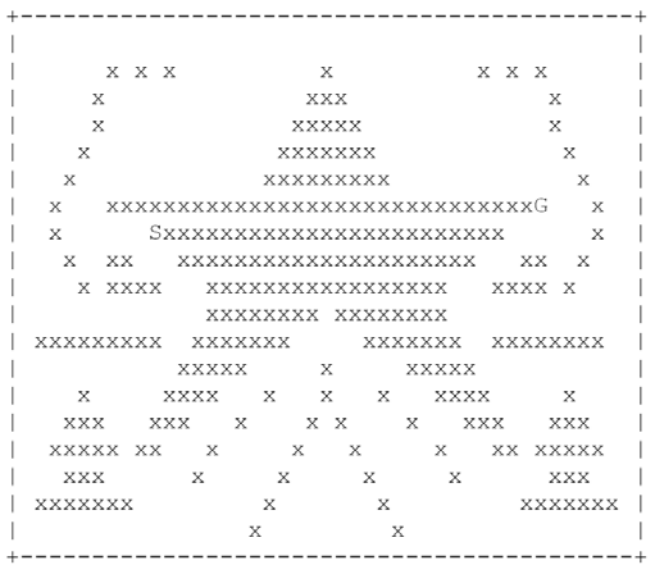


```
Number of state expansions: 2469
Cost: 50
"Elapsed time: 7274.595117 msecs"
```

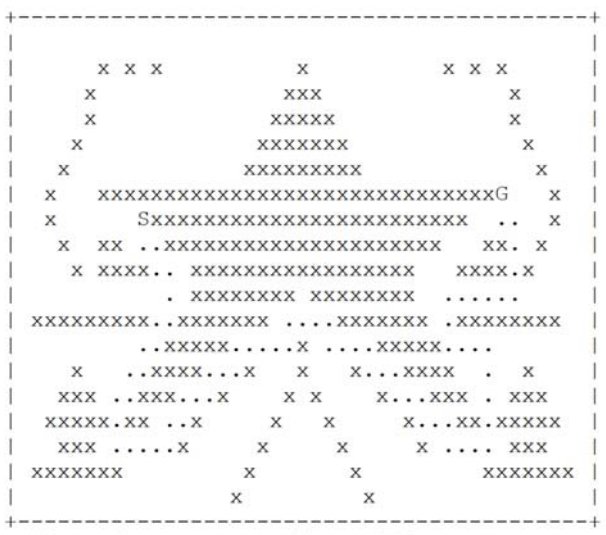
Map Three is where things got exciting. Both Best-First and A* outputted identical paths with a cost of 50 however A* was able to do this in almost a third of the time it took Best-First and with a fraction of the state expansions computed by Best-First. Best-First made 40721 expansions while A* only made 2496 expansions.

## MAP FOUR

This is the map that I created:

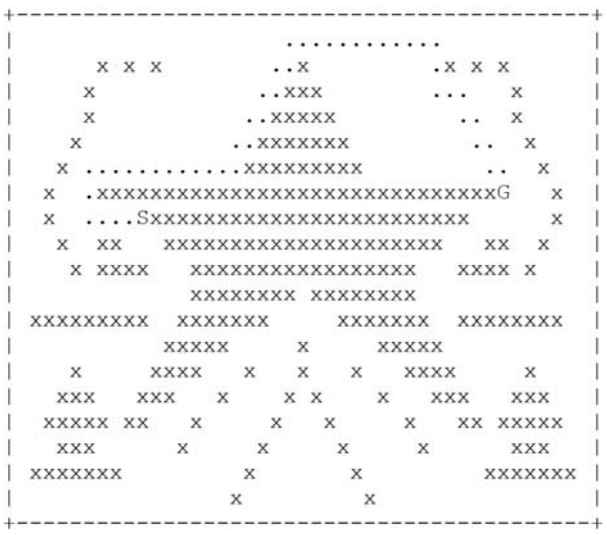


Best-First



```
Number of state expansions: 118
Cost: 72
"Elapsed time: 53.506 msecs"
```

A*



```
Number of state expansions: 742
Cost: 47
"Elapsed time: 1236.797388 msecs"
```

I placed multiple obstacles around the bottom half of the map in an attempt to prove that the Best-First algorithm finds a sub-optimal path from Start to Goal. The path produced by Best-First goes along the bottom, having to navigate around all of the obstacles thus resulting in a very long path over 1.5 times as long as the path produced by A*. It does not make many expansions (118) and as a result it does not take very long to compute (only 53msec) however the path is far from efficient. In comparison, A* makes many more state expansions (742) and takes a bit longer (1236msec) however the path found only has a length of 47 which is much better than the route found by Best-First as it also does not go from the bottom and instead chooses a path that does not wind between all the obstacles that the path chosen by Best-First encounters.

## CONCLUSION

Overall I have found that the A* algorithm finds a better or equally good path to the Best-First Algorithm. Most of the time it takes longer for A* to find a route as it computes many more expansions in comparison to Best-First however as I found with the map that I created, when there are a lot of obstacles involved in general direction where the heuristic is lowest, Best-First will still go in that direction since it only checks the heuristic, ignoring the cost of the path. As a result, Best-First can end up choosing a path that is not optimal.