

Comparison Between Greedy Best-First Search and A* Search

Introduction

This report will compare 2 algorithms A* search and Greedy best first search. Provided in the code snippet. Both algorithms will pathfinding graph traversal and their approach is somewhat different to find path.

KEY DIFFERENCES:

- Greedy Best First Search uses the function $f(n) = h(n)$ where $h(n)$ is a heuristic function. It considers only heuristic value ignoring the cost used to reach that node. While A* search uses function $f(n) = g(n) + h(n)$ where $g(n)$ is actual cost from start node to current node. $h(n)$ is heuristic value. The A* search dependent on both heuristic and actual path to the node
- Greedy Best first search is sometimes incomplete. It might not find a solution if it exists also. The A* search will find the path if the path exists.
- Greedy Best first search uses less memory compared to A* search because It will traverse only less heuristic values where as A* traverses more nodes to compare the actual costs.
- A* search finds optimal path whereas Greedy best first search does not guarantee to find optimal path.
- Complexity of A* is somewhat more than Greedy Best First Search.

Conclusion:

Both the algorithms have merits and demerits as discussed in the key differences section. A* is more robust and mostly finds the path. Greedy best first search is somewhat faster and doesn't guarantee the solution. Depending upon the scenario we have to choose the algorithm.