

Assignment No 2: Best-First search in Graph problem solving

CSE-0408 Summer 2021

Name: Nishu Akter

*Department of Computer Science and Engineering
State University of Bangladesh (SUB)*

Dhaka, Bangladesh

email: nishuakter309@gmail.com

I. INTRODUCTION

Best first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighbouring nodes. Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal.

ACKNOWLEDGMENT

I would like to thank my honourable **Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

II. VARIANTS OF BEST FIRST SEARCH

The two variants of Best First Search are Greedy Best First Search and A* Best First Search. Greedy BFS: Greedy best-first search algorithm always selects the path which appears best at that moment. It is the combination of depth-first search and breadth-first search algorithms. It uses the heuristic function and search. Best-first search allows us to take the advantages of both algorithms.

A* BFS: A* BFS algorithm is a searching algorithm that searches for the shortest path between the initial and the final state. It is used in various applications, such as maps. In maps the A* BFS algorithm is used to calculate the shortest distance between the source (initial state) and the destination (final state).

III. ALGORITHM FOR BFS

Let S be the root/starting node of the graph.

Step 1: Start with node S and enqueue it to the queue.

Step 2: Repeat the following steps for all the nodes in the graph.

Step 3: Dequeue S and process it.

Step 4: Enqueue all the adjacent nodes of S and process them. [END OF LOOP]

Step 6: EXIT

IV. CONCLUSION

The breadth-first search technique is a method that is used to traverse all the nodes of a graph or a tree in a breadth-wise manner.

This technique is mostly used to find the shortest path between the nodes of a graph or in applications that require us to visit every adjacent node like in networks.