

BEST FIRST SEARCH ALGORITHM

AIM

To implement Best First Search algorithm using Python

ALGORITHM

1. Initialize priority queue `queue` with start node and cost 0, and an empty set `visited`.
2. While `queue` is not empty, retrieve lowest cost node and path.
3. If current node is goal, return path.
4. Mark current node as visited.
5. For each unvisited neighbor, extend path and add to `queue` with priority based on visited nodes.
6. If goal not found, return None.

CODE

```
from queue import PriorityQueue
def best_first_search(graph, start, goal):
    queue = PriorityQueue()
    queue.put((0, [start]))
    visited = set()
    while not queue.empty():
        _, path = queue.get()
        current_node = path[-1]
        if current_node == goal:
            return path
        visited.add(current_node)
        for neighbor in graph.get(current_node, []):
            if neighbor not in visited:
                new_path = path + [neighbor]
                queue.put((len(visited), new_path))
    return None

if __name__ == "__main__":
    graph = {'A': ['B', 'C'], 'B': ['D', 'E'], 'C': ['F'], 'D': [], 'E': [], 'F': []}
    start, goal = 'A', 'F'
```

```
path = best_first_search(graph, start, goal)
if path:
    print("Path exists:", path)
else:
    print("Path does not exist.")
```

OUTPUT

```
= RESTART: C:/Users/Saaniya/Downloads/ai/10.py
Path exists: ['A', 'C', 'F']
|
```