

1. Write a Program in Prolog to solve any problem using Best First Search.

Answer:

% Define your graph with weighted edges and heuristics.

% Replace these with your actual graph and heuristics.

edge(a, b, 2).

edge(b, c, 3).

edge(b, d, 4).

edge(c, e, 5).

edge(d, e, 1).

edge(e, f, 2).

% Define heuristic values (straight-line distances to the goal).

heuristic(a, 6). % Replace with your specific values.

heuristic(b, 5).

heuristic(c, 4).

heuristic(d, 3).

heuristic(e, 2).

heuristic(f, 0).

% Define a predicate to calculate the total estimated cost.

total\_cost(Node, Path, Cost) :-

path\_cost(Path, PCost),

heuristic(Node, HCost),

Cost is PCost + HCost.

path\_cost([], 0).

path\_cost([A, B | Tail], Cost) :-

edge(A, B, EdgeCost),

path\_cost([B | Tail], RestCost),

Cost is EdgeCost + RestCost.

% Define Best-First Search algorithm.

best\_first\_search(Start, Goal, Path) :-

best\_first\_search\_internal([ [Start] ], Goal, RevPath),

reverse(RevPath, Path).

best\_first\_search\_internal([ [Goal | Path] | \_ ], Goal, [Goal | Path]).

best\_first\_search\_internal([ [Node | Path] | Rest ], Goal, Result) :-

findall([Next, Node | Path], (edge(Node, Next, \_), not(member(Next, Path))),

NewPaths),

append(Rest, NewPaths, AllPaths),

predsort(compare, AllPaths, SortedPaths),

best\_first\_search\_internal(SortedPaths, Goal, Result).

% Comparison function for sorting paths based on total cost.

compare(Result, [\_, \_, Path1], [\_, \_, Path2]) :-

total\_cost(Path1, Path1, Cost1),

total\_cost(Path2, Path2, Cost2),

compare\_paths(Cost1, Cost2, Result).

```
compare_paths(Cost1, Cost2, Result) :-
    (Cost1 < Cost2 -> Result = (<);
    Cost1 > Cost2 -> Result = (>);
    Result = (=)).
```

% Example usage:

% To find the path, call best\_first\_search(StartNode, GoalNode, Path).

% Replace StartNode and GoalNode with your problem's specific values.

Output:

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license, for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic), or ?- apropos(Word).

?- consult('C:/Users/ashis/OneDrive/Documents/(NCER) final year/sem 7/college authority or teacher given thing and my respective work/AI/AI practical/3/bfs.pl').
ERROR: c:/users/ashis/onedrive/documents/(ncer) final year/sem 7/college authority or teacher given thing and my respective work/ai/ai practical/3/bfs.pl
:43:
ERROR: No permission to modify static procedure 'compare/3'
true.

?- best_first_search(a, f, Path).
Path = [a, b, c, e, f].

?- best_first_search(a, c, Path).
Path = [a, b, c].

?-
```

2. Write a Program in Python to solve any problem using Best First Search.

Answer:

```
from queue import PriorityQueue
```

```
def best_first_search(graph, start, goal):
    frontier = PriorityQueue()
    frontier.put(start) # Use a priority queue with the initial node
    came_from = {} # Dictionary to store the best path
    came_from[start] = None
```

```
    while not frontier.empty():
        current = frontier.get()
        if current == goal:
            return reconstruct_path(came_from, current)
```

```
        for neighbor, weight in graph.get(current, []):
            if neighbor not in came_from:
                came_from[neighbor] = current
                frontier.put(neighbor)
```

```
    return None
```

```
def reconstruct_path(came_from, current):
```

```

path = []
while current:
    path.insert(0, current)
    current = came_from[current]
return path

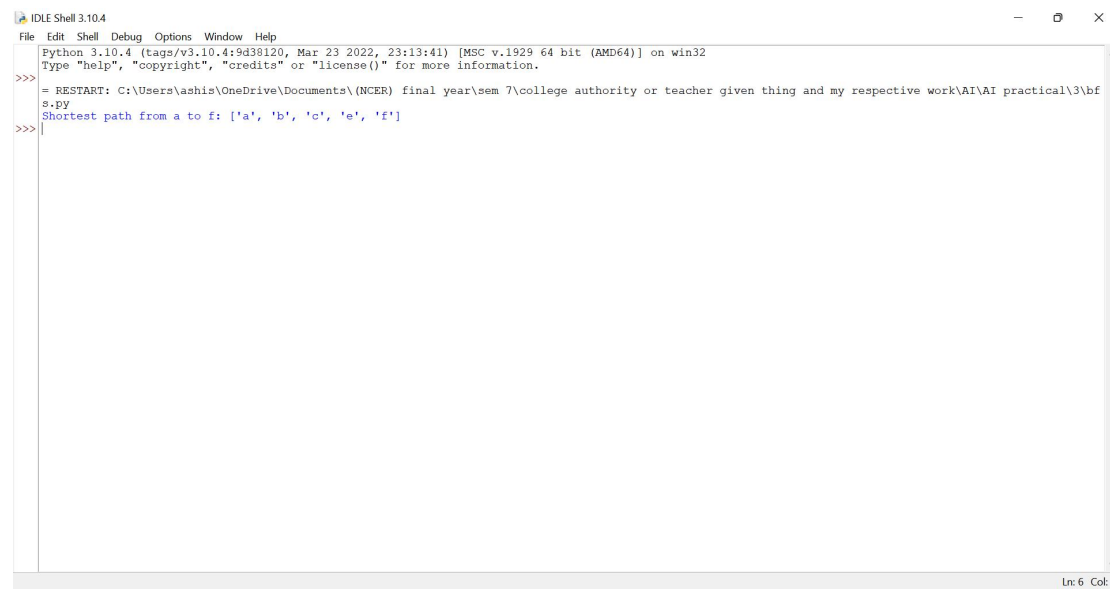
# Example usage:
graph = {
    'a': [('b', 2)],
    'b': [('c', 3), ('d', 4)],
    'c': [('e', 5)],
    'd': [('e', 1)],
    'e': [('f', 2)],
}

start_node = 'a'
goal_node = 'f'

path = best_first_search(graph, start_node, goal_node)
if path:
    print(f"Shortest path from {start_node} to {goal_node}: {path}")
else:
    print(f"No path found from {start_node} to {goal_node}")

```

Output:



```

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ashis\OneDrive\Documents\NCER final year\sem 7\college authority or teacher given thing and my respective work\AI\AI practical\3\bf
s.py
Shortest path from a to f: ['a', 'b', 'c', 'e', 'f']
>>>

```