% ----------------------------------------

% Best First Search in Prolog (GNU Prolog Compatible)

% ----------------------------------------

% Graph representation: edge(Node, Neighbor, Cost).

% Heuristic values: heuristic(Node, HValue).

% ----------------------------------------

% Example Graph (you can edit as needed)

edge(a, b, 1).

edge(a, c, 1).

edge(b, d, 1).

edge(b, e, 1).

edge(c, f, 1).

edge(c, g, 1).

% Heuristic values (h-values) for each node

heuristic(a, 5).

heuristic(b, 4).

heuristic(c, 3).

heuristic(d, 2).

heuristic(e, 0). % Goal node

heuristic(f, 6).

heuristic(g, 1).

% ----------------------------------------

% Best First Search Algorithm

% ----------------------------------------

% best\_first\_search(Start, Goal, Path).

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

heuristic(Start, H),

bestfs([H-[Start]], Goal, RevPath),

reverse(RevPath, Path).

% If the first node is the goal

bestfs([\_-[Node|RestPath] | \_], Node, [Node|RestPath]).

% Otherwise, expand the best node

bestfs([\_-[Node|RestPath] | Rest], Goal, Path) :-

findall(H-[Next,Node|RestPath],

(edge(Node, Next, \_), heuristic(Next, H), \+ member(Next, [Node|RestPath])),

Children),

append(Rest, Children, NewOpen),

keysort(NewOpen, SortedOpen), % GNU Prolog supports this

bestfs(SortedOpen, Goal, Path).

% ----------------------------------------

% Demo Execution

% ----------------------------------------

:- initialization(main).

main :-

write('--- Best First Search Demo ---'), nl,

best\_first\_search(a, e, Path),

write('Path found: '), write(Path), nl,

halt.