Exp27 Write a Prolog Program to implement Best First Search algorithm

```
% Graph edges
edge(a,b).
edge(a,c).
edge(b,d).
edge(c,d).
edge(c,e).
edge(d,goal).
edge(e,goal).

% Heuristic values
h(a,6).
h(b,4).
h(c,2).
h(d,1).
h(e,3).
h(goal,0).

% Best First Search
bestfs(Start, Goal, Path) :-
   bfs([[Start]], Goal, RevPath),
   reverse(RevPath, Path).

bfs([[Goal|Rest]|_], Goal, [Goal|Rest]).
bfs([[Node|RestPath]|Others], Goal, Path) :-
   findall([Next,Node|RestPath],
         (edge(Node,Next), \+ member(Next,[Node|RestPath])),
         NewPaths),
   sort_paths(NewPaths, Sorted),
```

```
    append(Others, Sorted, Total),
    bfs(Total, Goal, Path).


% Sort paths by heuristic of head node
sort_paths(Paths, Sorted) :-
    map_list_to_pairs(
        [P,H]>>(P=[N|_],h(N,H)),
        Paths, Pairs),
    keysort(Pairs, SortedPairs),
    pairs_values(SortedPairs, Sorted).
```

```
% Updating index for library c:/users/administrator/downloads/swipl/library/
% Updating index for library c:/users/administrator/downloads/swipl/xpce/prolog/lib/
.

?- bestfs(a, goal, Path).
Path = [a, c, d, goal]
```