This assignment asks you to apply the A* search algorithm to the processing of propositional Prolog knowledge bases such as

\[
q:- a.
q:- b,c.
a:- d,e.
a:- c,e,f.
b:- c.
c:- e,f.
e.
f:- e.
\]

which we can represent as the list

\[
[[q,a],[q,b,c],[a,d,e],[a,c,e,f],[b,c],[c,e,f],[e],[f,e]]
\]

and use as KB in the clauses

\[
arc([H|T],Node,Cost,KB) :- member([H|B],KB), append(B,T,Node),
length(B,L), Cost is 1+ L/(L+1).
\]

\[
heuristic(Node,H) :- length(Node,H).
\]

\[
goal([]).
\]

Your task is to define the predicate

\[
astar(+Node,?Path,?Cost,+KB)
\]

that implements A*, returning a path from Node to the goal node [] with minimal cost, given KB. Test your code with queries such as

\[
?- astar([q],Path,Cost,
[[q,a],[q,b,c],[a,d,e],[a,c,e,f],[b,c],[c,e,f],[e],[f,e]]).
\]

Cost = 11.916666666666668,
Path = [[q], [a], [c, e, f], [e, f, e, f], [e, e, f],
[e, f], [f], [e], []] ;
Cost = 13.499999999999998,
Path = [[q], [b, c], [c, c], [e, f, c], [f, c], [e, c], [c],
[e, f], [f], [e], []] ;
false

?- astar([q], Path, Cost, [[q, a], [q, b, c], [a], [b], [c]]).
Cost = 2.5,
Path = [[q], [a], []];
Cost = 3.6666666666666665,
Path = [[q], [b, c], [c], []];
false

**Hint**  Modify the skeletal search algorithm

```prolog
search([Node|_]) :- goal(Node).
search([Node|More]) :- findall(X, arc(Node, X), Children),
          add2frontier(Children, More, New),
          search(New).
```

so that the head of the list `New` obtained in `add2frontier` has $f$-value no larger than any in `New`'s tail, where

$$f(node) = \text{cost}(node) + h(node).$$

Let the frontier be a list of path-cost pairs (instead of just nodes), being careful to update path cost, and to bring in the heuristic function in forming the frontier `New`

```prolog
less-than([Node1|_, Cost1], [Node2|_, Cost2]) :-
  heuristic(Node1, Hvalue1), heuristic(Node2, Hvalue2),
  F1 is Cost1+Hvalue1, F2 is Cost2+Hvalue2,
  F1 =< F2.
```