Introduction to Artificial Intelligence

Sample Answers to Lab - blind search*

1. The breadth-first search strategy can be obtained by extending `coreSearch.pl` with

```
choose(Path,[Path|Paths],Paths).
combine(Exps,Paths,NewPaths):-
        append(Paths,Exps,NewPaths).
```

2. The depth-first search strategy can be obtained by extending `coreSearch.pl` with

```
choose(Path,[Path|Paths],Paths).
combine(Exps,Paths,NewPaths):-
        append(Exps,Paths,NewPaths).
```

3.
```
| ?- [coreSearch,breadthFirst,testExample1].
| ?- search([[a]],X).
X = [a,f] ? ;
X = [a,b,f] ? ;
no
| ?- [coreSearch,depthFirst,testExample1].
| ?- search([[a]],X).
X = [a,b,f] ? ;
X = [a,f] ? ;
no
```


The two solutions are presented in different orders by the two search strategies.

4. The given implementation of depth-first search is very inefficient. Although the search space is traversed in the right way, at any given node the program stores a list of all children, then goes to work on the 'left-most' child (all of whose children will, in turn, be remembered). Where the branching factor is high, or where it is low but the depth searched is high, then there will be a lot of unnecessary remembering occuring. By exploiting Prolog's backtracking:

```
dfSearch([Node|Path],X):-
        goal(Node),
        reverse([Node|Path],X).

dfSearch([Node|Path],X):-
        NewNode expands [Node|Path],
        dfSearch([NewNode,Node|Path],X).

NewState expands [State|_]:-
        arc(State,NewState).
```

*Thanks to Keith Clark

5.
```
| ?- [coreSearch,breadthFirst,testExample2].
| ?- search([[a]],X).
X = [a,f] ? ;
X = [a,b,f] ? ;
X = [a,b,c,b,f] ? ;
X = [a,b,c,b,c,b,f] ? ;
X = [a,b,c,b,c,b,c,b,f] ?
...

| ?- [coreSearch,depthFirst,testExample2].
| ?- search([[a]],X).
! Resource error: insufficient memory
```

   Breadth-first search computes infinitely many solutions. Depth-first does not terminate, as it gets "stuck" in an infinite branch of the search space.

6. The definition of `search` in file `coreSearch.pl` can be modified as follows:

```
search(Paths,X):-
        choose([Node|Path],Paths,_),
        goal(Node),
        reverse([Node|Path],X).

search(Paths,Path):-
        choose(P,Paths,RestofPaths),
        findall([S|P],
                (S expands P, \+ member(S,P)),
                Exps),
        combine(Exps,RestofPaths,NewPaths),
        search(NewPaths,Path).
```


7. Assume the new definition for `search` is in file `loopCheckCoreSearch.pl`:

```
| ?- [loopCheckCoreSearch,breadthFirst,testExample2].
| ?- search([[a]],X).
X = [a,f] ? ;
X = [a,b,f] ? ;
no

| ?- [loopCheckCoreSearch,depthFirst,testExample2].
| ?- search([[a]],X).
X = [a,b,f] ? ;
X = [a,f] ? ;
no
```