```
member(a, [a, b, c]).
length([a, b, c], 3).
append([a, b], [c, d], [a, b, c, d]).
sort([c, b, a], [a, b, c]).
```

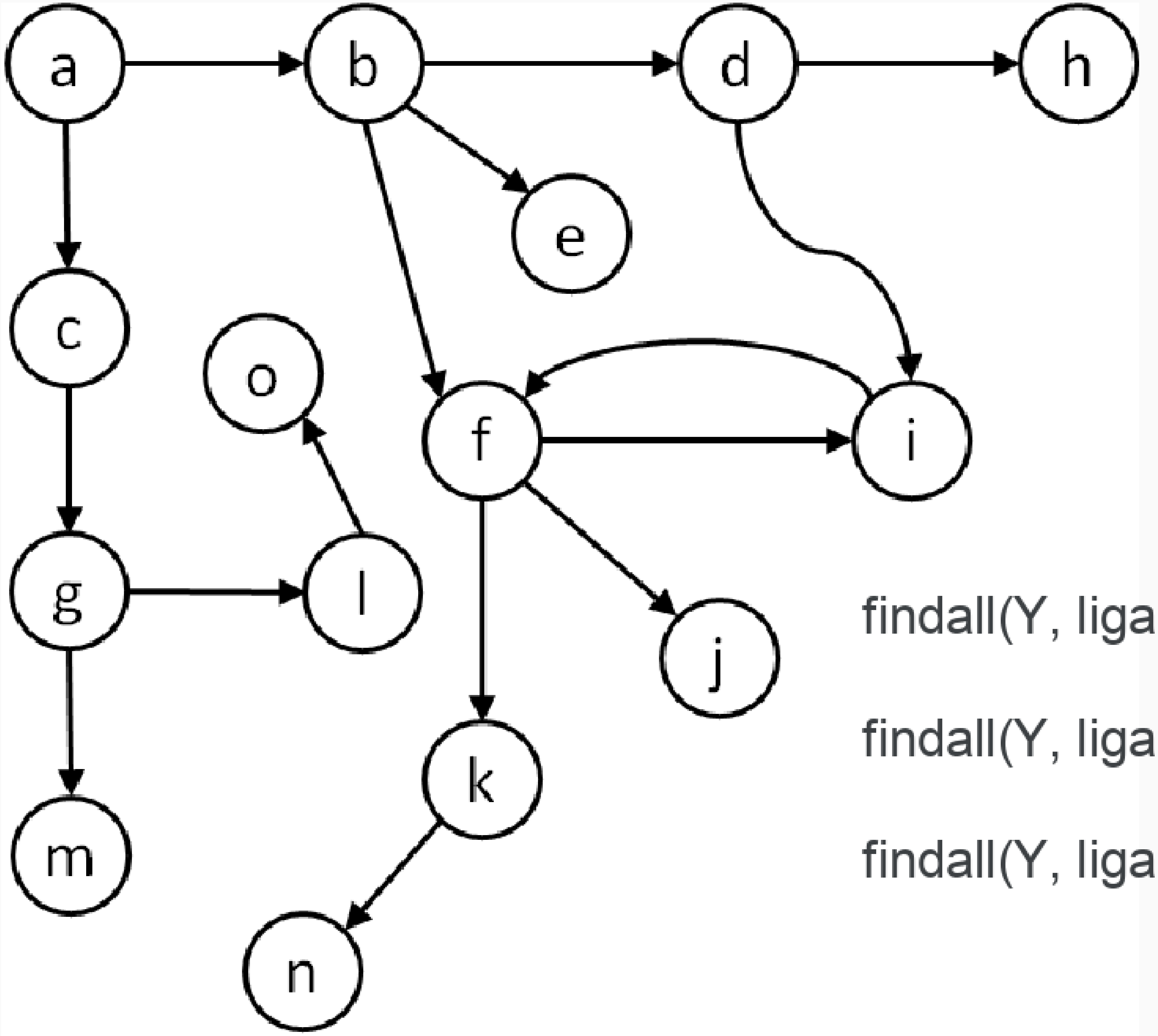
```
:-include('file.pl').
:-load_files('file.pl').
:-ensure_loaded('file.pl').
```

```
:- use_module(library(lists)).            10.23
```

```
nth0/3 /4
nth1/3 /4
select/3 / 4
reverse/2.
append/2
delete/3 /4
last/2 /3
transpose/2
maplist/2 /3 /4
remove_dups/2
sumlist/2
```

```
nth0(2, [a, b, c], c).
nth1(2, [a, b, c], b).

nth0(1, [a, b, c], b, [a, c]).
nth1(1, [a, b, c], a, [b, c]).

select(b, [a, b, c], [a, c]).

select(b, [a, b, c], d, [a, d, c]).
```

```
delete([a, b, a, c, a], a, [b, c]).
delete([a, b, a, c, a], a, 2, [b, c, a]).

last([a, b, c], c).
last([a, b], c, [a, b, c]).

remove_dups([a, b, a, c, a], [a, b, c])

sumlist([1, 3, 5], 9).
```



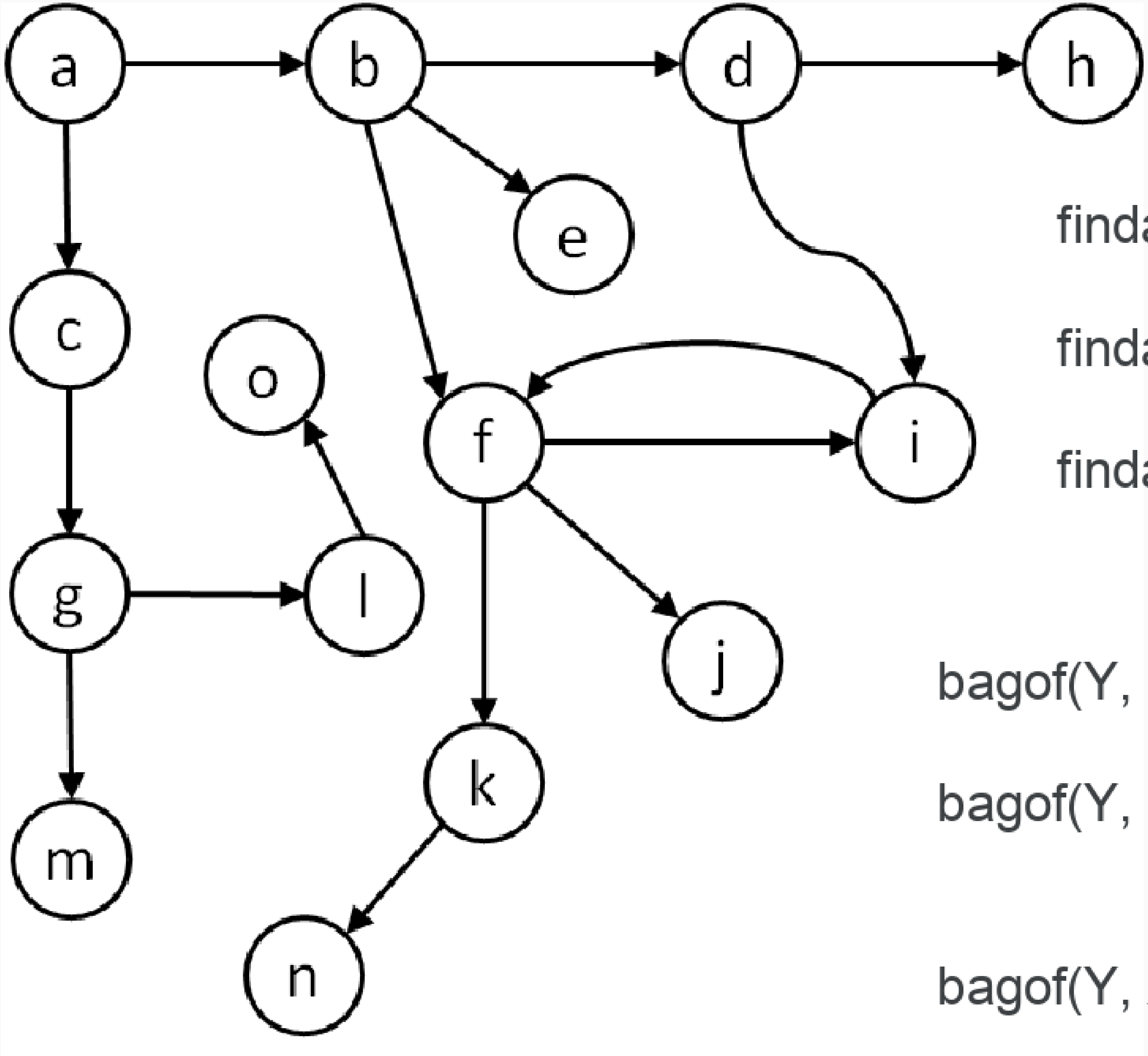
findall
bagof
setof

findall(+Var, +Goal, -List).

findall(Y, ligado(a, Y), L). L = [b, c]

findall(Y, ligado(X, Y), L). L = [b,c,d,e,f,g,h,i,i,j,k,l,m,n,o,

findall(Y, ligado(e, Y), L). L = [ ]

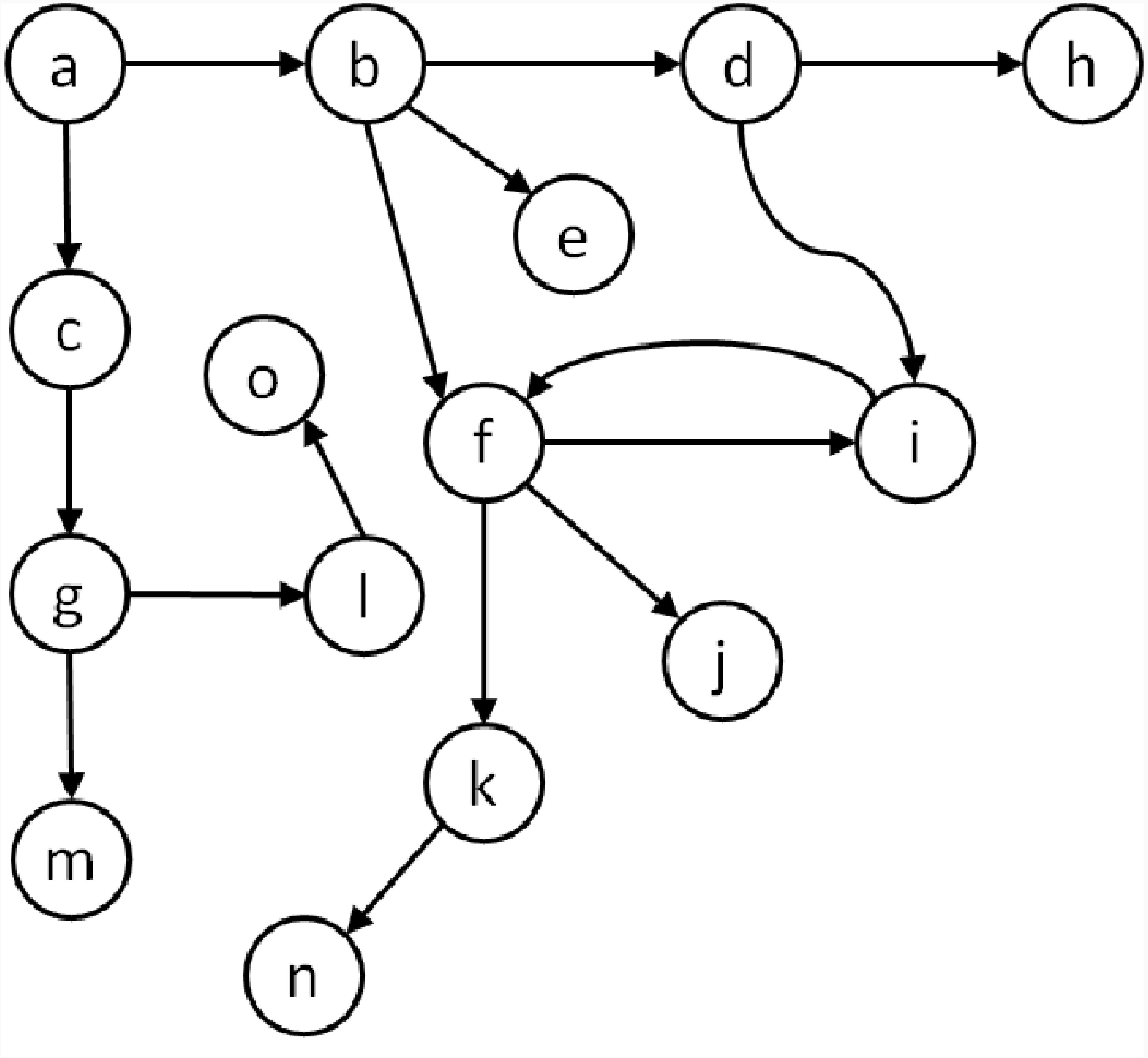


```
findall(Y, ligado(a, Y), L).      L = [b, c]

findall(Y, ligado(X, Y), L).      L = [b,c,d,e,f,g,h,i,i,j,.

findall(Y, ligado(e, Y), L).       L = [ ]


bagof(Y, ligado(a, Y), L).          L = [b, c]

bagof(Y, ligado(X, Y), L).          X = a, L = [b,c]  ;
                                       X = b, L = [d, e, f]  ; ...

bagof(Y, X^ligado(X, Y), L).     L = [b,c,d,e,f,g,h,i,i,j,...]

bagof(Y, ligado(e, Y), L).          no
```

```
bagof(Z, X^obj(X, Y, Z), L).
```

```
findall(X-Y, validMove(Tab, X-Y, NTab), List).

setof(Val-X-Y, ( validMove(Tab, X-Y, NTab), value(NTab, Val) ), [VM-Xi-Yi | _] ).
```

1.a), 1.b), 6



```
prof(End, End, [End]).

prof(Start, End, [Start|Path]):-
    Start\= End,
    ligado(Start, Mid),
    prof(Mid, End, Path).
```

```
prof(Start, End, Path):-
    prof(Start, End, [Start], Path).

prof(End, End, Path, Path).
prof(Start, End, Temp, Path):-
    ligado(Start, Next),
    \+ member(Next, Temp),
    append(Temp, [Next], NTemp),
    prof(Next, End, NTemp, Path).
```