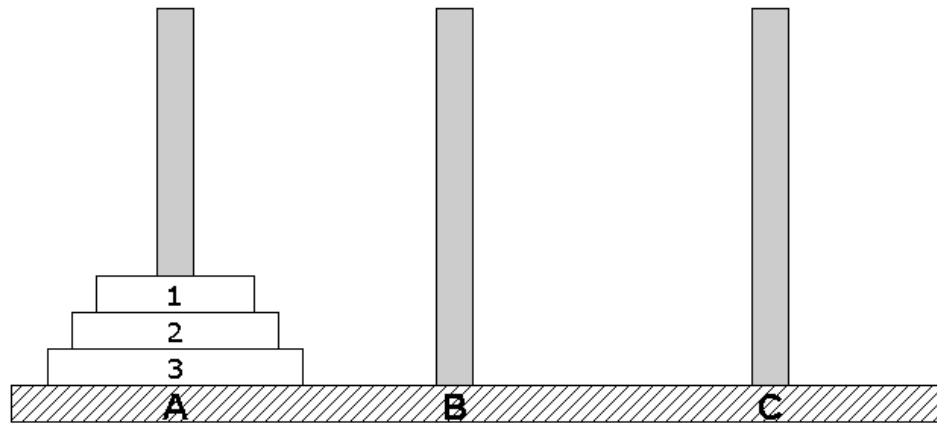


Recherche en profondeur

```
resoudre(N,[N]) :- but(N).  
resoudre(N,[N | Solution]) :-  
    successeur(N,Nsuivant), resoudre(Nsuivant,Solution).
```

Il faut donc définir le but et définir les noeuds successeurs.

Les tours de Hanoi



État initial: $[A,B,C] = [[1,2,3],[],[]]$

but($[[[]],[],_]$).

Les tours de Hanoi (mouvements)

legal(_,[]).

legal(D1,[D2|_]):- D1<D2.

successeur([[T|L1],L2,L3],[L1,[T|L2],L3]) :- legal(T,L2).

successeur([[T|L1],L2,L3],[L1,L2,[T|L3]]) :- legal(T,L3).

successeur([L1,[T|L2],L3],[[T|L1],L2,L3]) :- legal(T,L1).

successeur([L1,[T|L2],L3],[L1,L2,[T|L3]]) :- legal(T,L3).

successeur([L1,L2,[T|L3]],[[T|L1],L2,L3]) :- legal(T,L1).

successeur([L1,L2,[T|L3]],[L1,[T|L2],L3]) :- legal(T,L2).

Les tours de Hanoi (version 1)

```
but([],[],_).  
resoudre(E,[E]) :- but(E).  
resoudre(E,[E|Solution]) :- successeur(E,Esuivant),  
                             resoudre(Esuivant,Solution).
```

?!

Les tours de Hanoi (version 2)

```
resoudre(Noeud, Solution) :- profondeur([], Noeud, Solution),  
                             montre(Solution).
```

```
profondeur(Chemin, Noeud,[Noeud | Chemin]) :- but(Noeud).
```

```
profondeur(Chemin, Noeud, Solution) :-
```

```
    successeur(Noeud, Nsuivant),
```

```
    \+member(Nsuivant, Chemin),
```

```
    profondeur([Nsuivant | Chemin], Nsuivant, Solution).
```

```
voir([]).
```

```
voir([Noeud|Chemin]) :- voir(Chemin), nl, write(Noeud).
```

Les tours de Hanoi (version 3)

```
resoudre(E,[E],_) :- but(E).  
resoudre(E,[E|Solution],Pmax) :- Pmax>0,  
    successeur(E,Esuivant),  
    Pmax1 is Pmax-1,  
    resoudre(Esuivant,Solution,Pmax1),  
    write(Esuivant), nl.
```

Recherche en largeur

```
resoudre(Racine, Solution):- largeur ([[Racine]],Solution).
% largeur(liste de chemin, solution)
largeur([[Noeud | Chemin] | _],[Noeud | Chemin]):- but(Noeud).
largeur([Chemin | Chemins], Solution):-
    etendre(Chemin,NChemins),
    append(Chemins,NChemins,Chemins1),
    largeur(Chemins1,Solution).
etendre([Noeud | Chemin],NChemins):-
    bagof([NNoeud,Noeud | Chemin],
        (successeur(Noeud,NNoeud),
         \+member(NNoeud,[Noeud | Chemin])),
        NChemins),
    !.
etendre(Chemin,[]).
```

Les tours de Hanoi (version 4)

?- successeur([[1,2,3],[],[[]],S).

S = [[2, 3], [1], []] ;

S = [[2, 3], [], [1]] ;

?- etendre([[[1,2,3],[],[[]]],S).

S = [[[[2, 3], [1], []], [[1, 2, 3], [], []]], [[2, 3], [], [1]], [[1, 2, 3], [], []]].

Les tours de Hanoi (version 4)

```
resoudre(Racine, Solution):- largeur([[Racine]],Solution),  
                             voir(Solution).
```

```
voir([]).
```

```
voir([Noeud|Chemin]) :- voir(Chemin), nl, write(Noeud). .
```

```
?- resoudre([[1,2,3],[[],[]],S).
```

```
[[1,2,3],[[],[]]
```

```
[[2,3],[[],[1]]
```

```
[[3],[2],[1]]
```

```
[[3],[1,2],[[]]
```

```
[[[]],[1,2],[3]]
```

```
[[1],[2],[3]]
```

```
[[1],[[]],[2,3]]
```

```
[[[]],[[]],[1,2,3]]
```