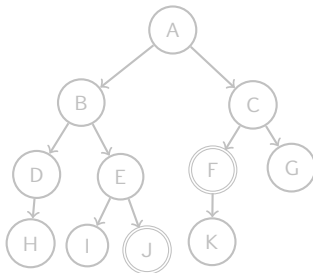

```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
-	-	-	-	{{A}}



```
Ch <- Premier(Liste); n <- Faux;
```

```
while Ch!= et non(n) do
```

```
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
```

```
n1 <- successeur(n);
```

```
while non(n) et n1 est valide do
```

```
  if n1 est solution then
```

```
    Sol Ch Uf n1; n <- Vrai;
```

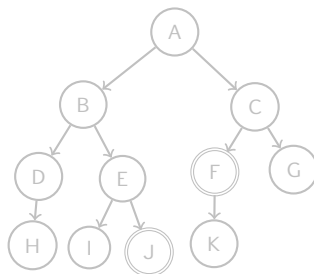
```
else Liste <- Liste Uf (Ch Uf n1);
```

```
n1 <- successeur(n);
```

```
Ch Premier(Liste);
```

```
return n;
```

Ch	n	n1	fin	Liste
{A}	-	-	Faux	{{A}}

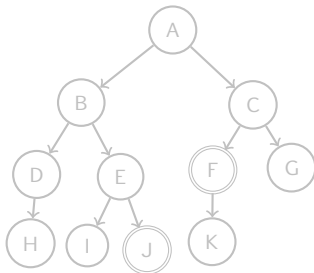


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	-	-	Faux	{{A}}

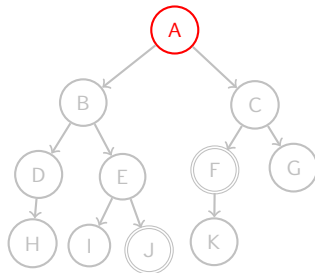


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	-	Faux	{}

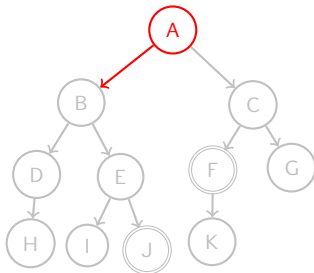


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	B	Faux	{}

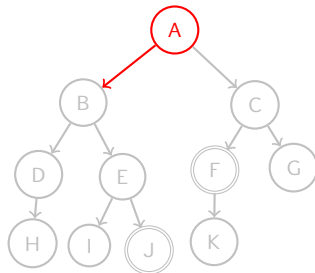


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	B	Faux	{}

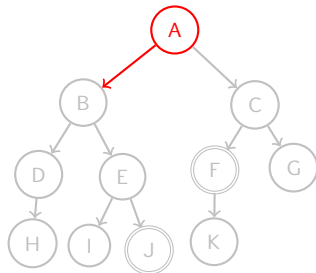


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	B	Faux	{}

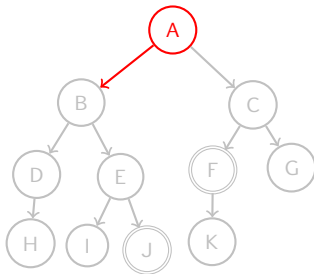


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	B	Faux	{{AB}}

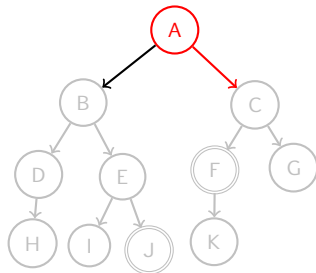


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	C	Faux	{{AB}}

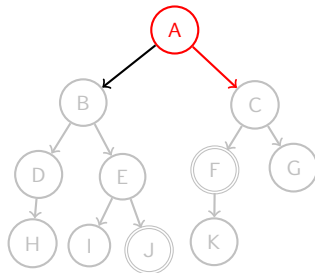


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	C	Faux	{{AB}}

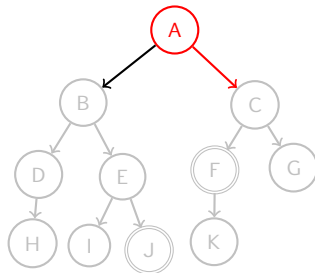


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	C	Faux	{{AB}}

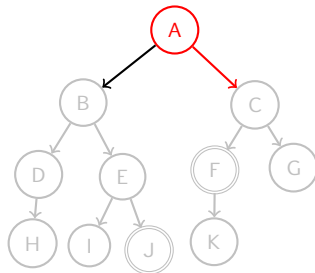


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A	C	Faux	{{AB},{AC}}

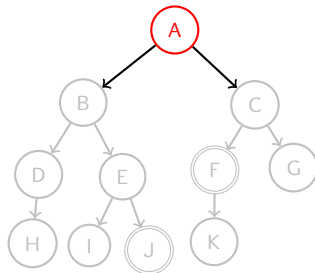


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}

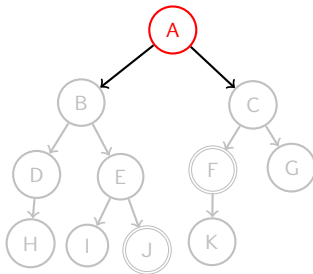


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}

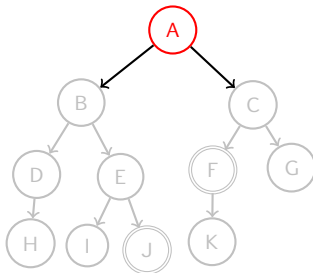


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}				

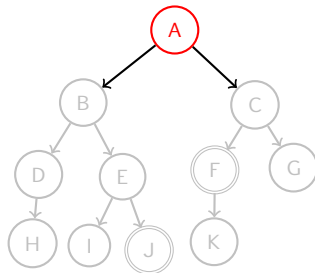


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}				

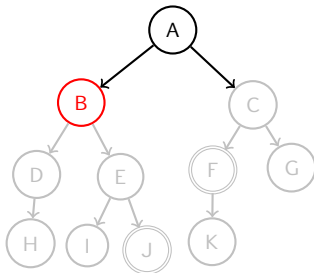



```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{AC}

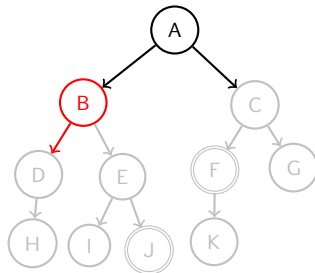


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
  n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
  n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	D		{AC}

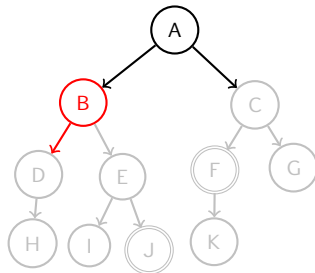


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
  n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	D		{AC}

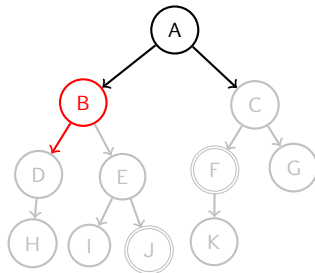


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	D		{AC}

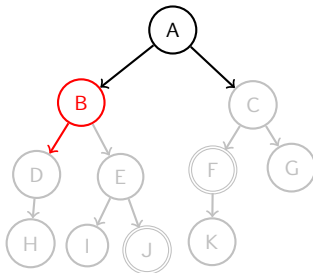


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	D		{{AC},{ABD}}

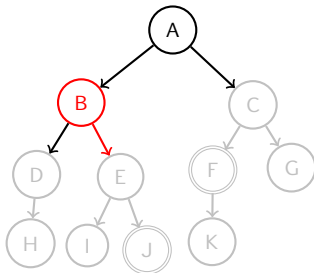


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	E		{{AC},{ABD}}

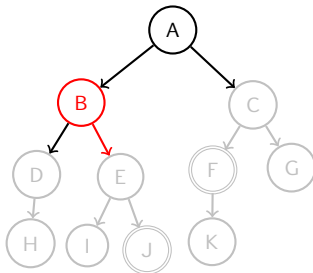


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	E		{{AC},{ABD}}

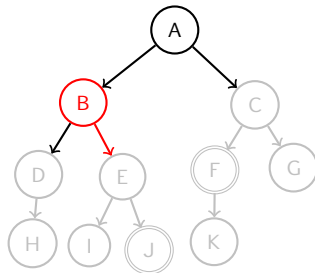


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	E		{{AC},{ABD}}

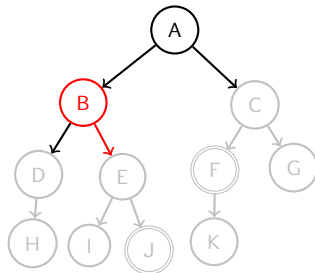


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B	E		{{AC},{ABD},{ABE}}

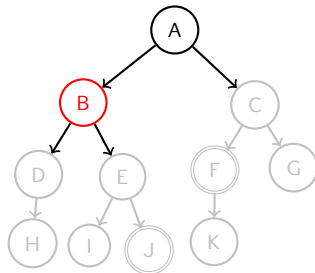


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}

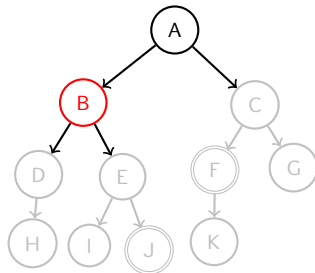


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
  n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}

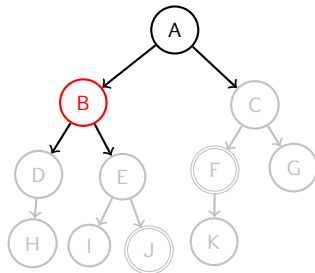


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
  n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}				

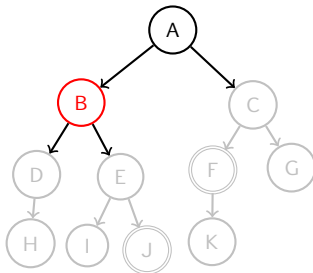


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}				

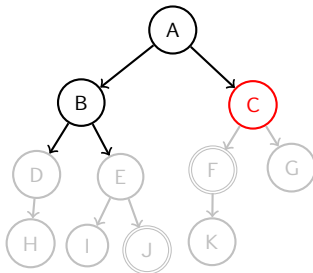


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C			{{ABD},{ABE}}

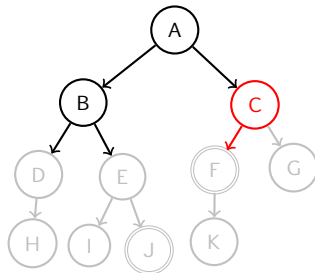


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	F		{{ABD},{ABE}}

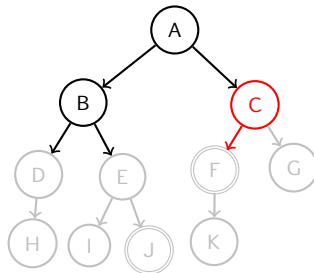


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	F		{{ABD},{ABE}}

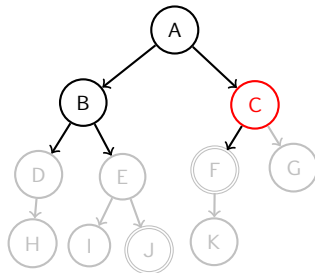


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	F		{{ABD},{ABE}}

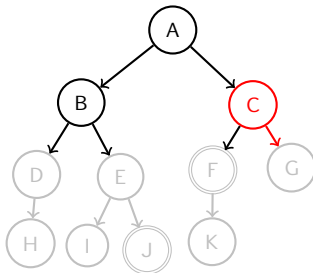


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	F	Vrai	{{ABD},{ABE}}

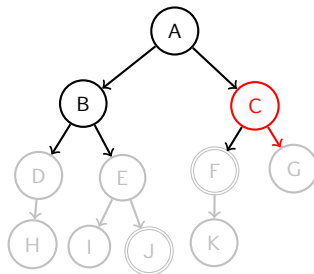


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}

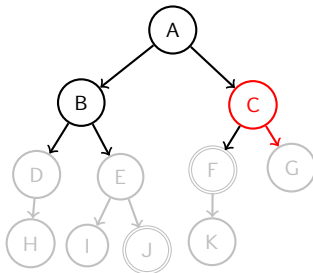


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}

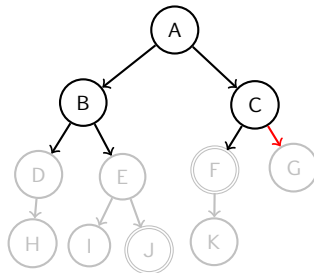


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}

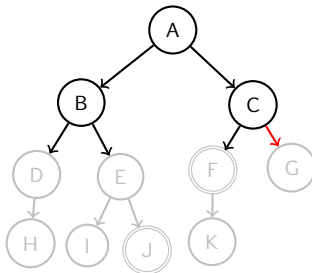


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}

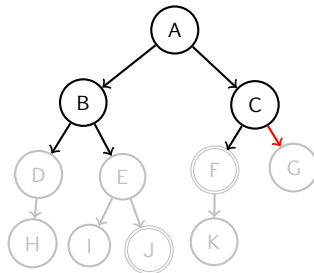


```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}



```

Ch <- Premier(Liste); n <- Faux;
while Ch!= et non(n) do
  Liste <- Liste Ch; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(n) et n1 est valide do
  if n1 est solution then
    Sol Ch Uf n1; n <- Vrai;
  else Liste <- Liste Uf (Ch Uf n1);
n1 <- successeur(n);
Ch Premier(Liste);
return n;

```

Ch	n	n1	fin	Liste
{A}	A		Faux	{{AB},{AC}}
{AB}	B			{{AC},{ABD},{ABE}}
{AC}	C	G	Vrai	{{ABD},{ABE}}

