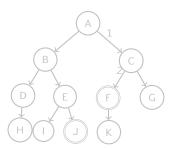
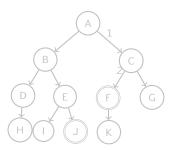
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf {n1}; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf {n1})}; n1 <- successeur(n); Ch Premier(Liste);

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



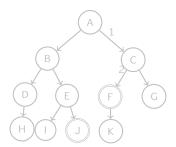
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste);

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



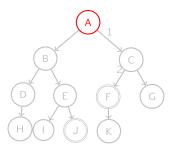
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

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



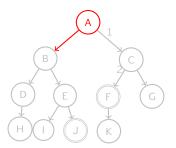
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

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



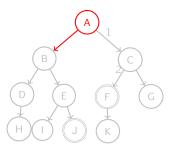
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste); return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

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



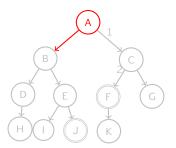
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
return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

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



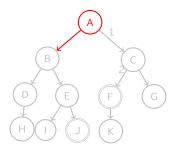
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {| Ch Uf [n1] |};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



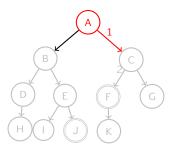
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {|Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{{AB}}	



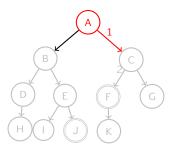
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);
Ch Premier(Liste);
return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

	Ch	n	n1	fin	Liste	Sol
ĺ	{A}	Α	С	Faux	{{AB}}	



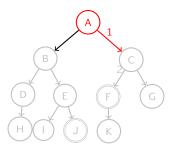
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	С	Faux	{{AB}}	



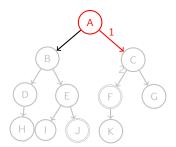
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {| Ch Uf [n1] |};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	С	Faux	{{AB}}	



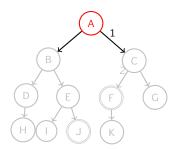
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {|Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

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



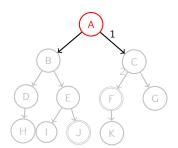
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

	Ch	n	n1	fin	Liste	Sol
	{A}	Α	1	Faux	{{AB},{AC}}	
1						



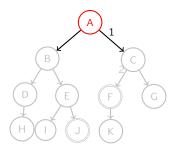
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {| Ch Uf [n1] |};
n1 <- successeur(n);
Ch Premier(Liste);

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



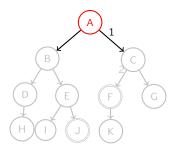
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}					



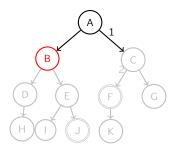
```
Ch <- Premier(Liste); fin <- Faux;
white Ch!= || et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);
Ch Premier(Liste);
```

Ch	n	n1	fin	Liste	Sol
{A} {AB}	Α	Τ.	Faux	{{AB},{AC}}	



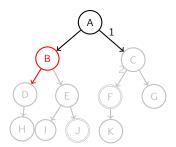
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
 Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
 n1 <- successeur(n);
while non(fin) et n1 est valide do
 if n1 est solution then
 Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {|Ch Uf [n1])};
 n1 <- successeur(n);
 Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В			{AC}	



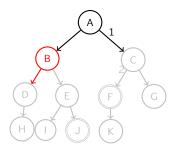
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste); return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

Ch	n	n1	fin	Liste	Sol
{A}	A	H (Faux	{{AB},{AC}}	
{AB}	В	D		{AC}	



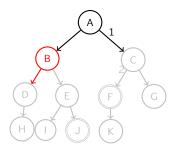
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	D		{AC}	



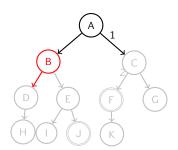
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {{Ch Uf [n1]}};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	D		{AC}	



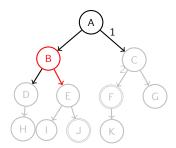
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

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



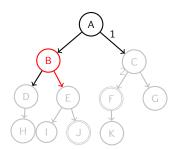
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
return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

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



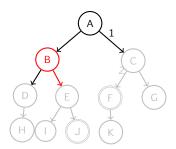
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
```

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



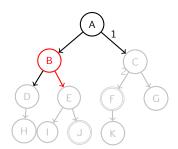
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	E		{{AC},{ABD}}	



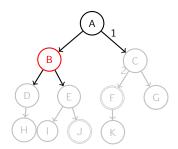
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {|Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

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



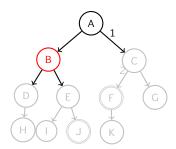
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	<u></u>	Faux	{{AB},{AC}} {{AC},{ABD},{ABE}}	



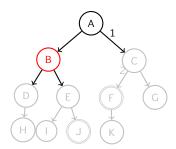
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf {n1}; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf {n1})}; n1 <- successeur(n); Ch Premier(Liste);
```

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



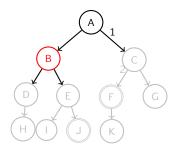
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
return fin;
```

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



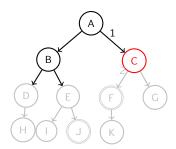
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
return fin;
```

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



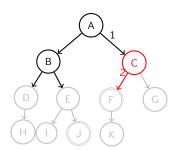
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste); return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

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С			{{ABD},{ABE}}	



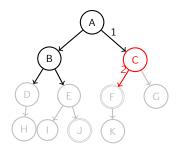
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste); return fin;
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



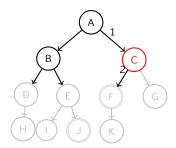
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf |n1|; fin <- Vrai; else Liste <- Liste Uf {|Ch Uf |n1}); n1 <- successeur(n); Ch Premier(Liste);
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	Τ	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



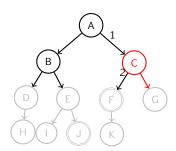
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n); Ch Premier(Liste);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

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



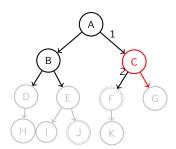
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf [n1])};
n1 <- successeur(n);
Ch Premier(Liste);
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	Τ	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F	Vrai	{{ABD},{ABE}}	$\{A,C,F\}$



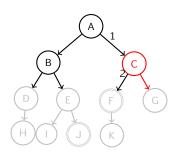
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste);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

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



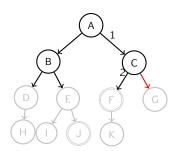
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / {Ch}; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste); return fin;
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



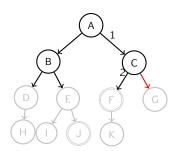
```
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste);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

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



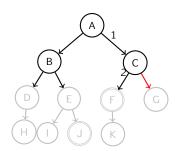
```
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);
Ch Premier(Liste);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

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf {n1}; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf {n1})}; n1 <- successeur(n); Ch Premier(Liste); return fin;
```

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



```
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf {n1}; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf {n1})); n1 <- successeur(n); Ch Premier(Liste); return fin;
```

1	Ch	n	n1	fin	Liste	Sol
ĺ	{A}	Α	1	Faux	{{AB},{AC}}	
ı	{AB}	В	1		{{AC},{ABD},{ABE}}	
İ	{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}

