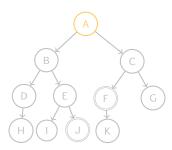
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch UF [n1]; fin <- Vrai; else Liste <- Liste UF [(Ch UF [n1])]; n1 <- successeur(n); Ch Premier(Liste);

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

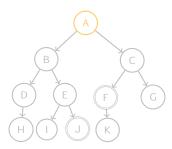
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow 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

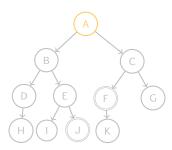
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

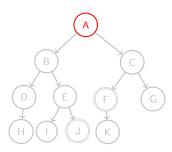
Ch Premier(Liste);

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



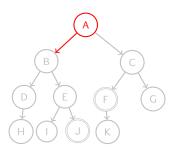
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf |n1|; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf |n1|)}; n1 <- successeur(n); Ch Premier(Liste);

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);

С	h	n	n1	fin	Liste	Sol
{/	\ }	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

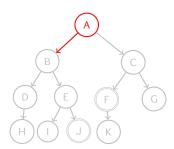
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

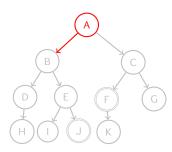
Ch Premier(Liste);

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);

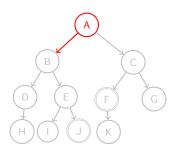
ſ	Ch	n	n1	fin	Liste	Sol
Ī	{A}	Α	В	Faux	{}	



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|));

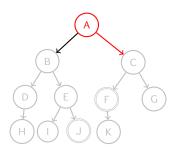
n1 <- successeur(n);
Ch Premier(Liste);
return fin;</pre>

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);

ſ	Ch	n	n1	fin	Liste	Sol
ĺ	{A}	Α	С	Faux	{{AB}}	
1						



$$\label{eq:ch} \begin{split} & \text{Ch} < \text{-} \, \text{Premier}(\text{Liste}); \text{fin} < \text{-} \, \text{Faux}; \\ & \text{while} \, \, \text{Ch}! = \{\} \, \text{et non}(\text{fin}) \, \text{do} \end{split}$$

 $Liste <- Liste \ / \ \{Ch\}; n <- dernierNoeud(Ch);$

n1 <- successeur(n);

while non(fin) et n1 est valide do

if n1 est solution then

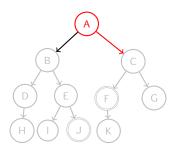
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

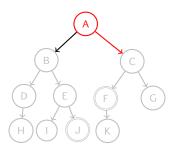
Ch Premier(Liste);

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);

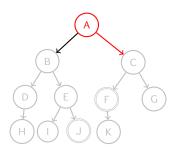
Ch	n	n1	fin	Liste	Sol
{A}	Α	С	Faux	{{AB}}	



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;

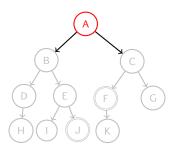
else Liste <- Liste Uf {(Ch Uf {n1}));
n1 <- successeur(n);
Ch Premier(Liste);
return fin;</pre>

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	n	n1	fin	Liste	Sol
{A}	Α	Т	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 n! est valide do

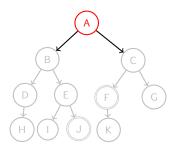
while non(fin) et n1 est valide do if n1 est solution then

Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste <- Liste Uf $\{(Ch\ Uf\ \{n1\})\};$

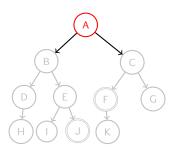
n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	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);

ſ	Ch	n	n1	fin	Liste	Sol
	{A} {AB}	Α		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

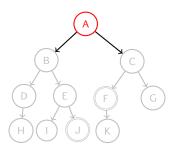
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste <- Liste Uf {(Ch Uf {n1})};

n1 <- successeur(n);

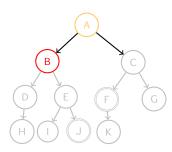
Ch Premier(Liste);

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



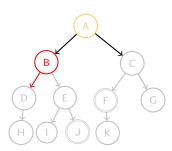
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste);

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



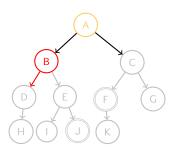
Ch <- Premier(Liste); fin <- Faux; while Ch!= [] et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n);

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



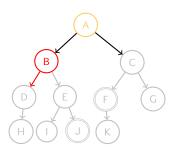
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);

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



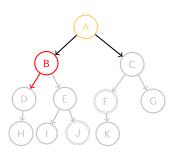
Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

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



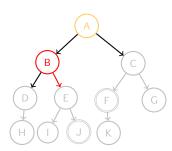
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);

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



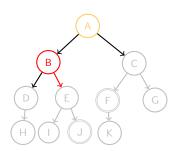
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

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



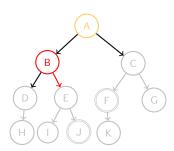
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do}
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);

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



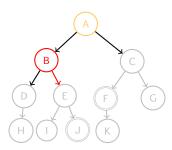
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1}));
n1 <- successeur(n);

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



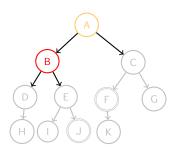
Ch <- Premier(Liste); fin <- Faux;
while Ch!= {| et non(fin) do
Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf {n1}; fin <- Vrai;
else Liste <- Liste Uf {(Ch Uf {n1})};
n1 <- successeur(n);

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



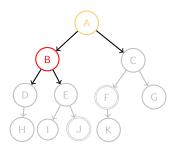
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

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



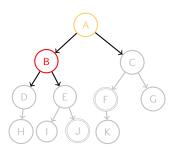
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

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

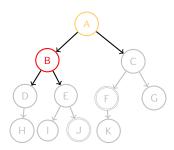
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

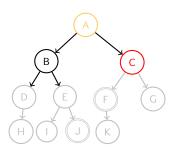
Ch Premier(Liste);

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



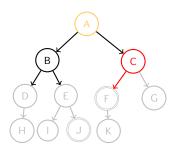
Ch <- Premier(Liste); fin <- Faux; while Ch!= [] et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n);

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



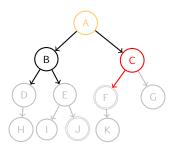
Ch <- Premier(Liste); fin <- Faux; while Ch!= [] et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n);

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



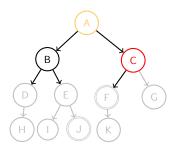
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

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



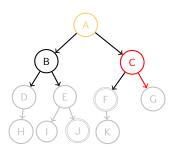
Ch <- Premier(Liste); fin <- Faux; while Ch!= [] et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n);

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



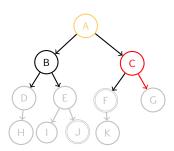
Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

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



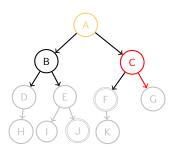
Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

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



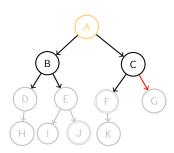
Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

Ch	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	n	n1	fin	Liste	Sol
{A}	Α		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);</pre>

n1 <- successeur(n):

while non(fin) et n1 est valide do

if n1 est solution then

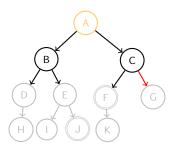
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste <- Liste Uf {(Ch Uf {n1})};

n1 <- successeur(n);

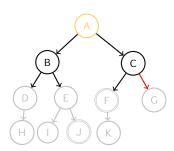
Ch Premier(Liste);

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



$$\label{eq:Characteristics} \begin{split} &\text{Ch} <\text{- Premier(Liste)}; &\text{fin} <\text{- Faux}; \\ &\text{while Ch!= }[] \text{ et non(fin) do} \\ &\text{Liste } <\text{- Liste } / [\text{Ch}]; n <\text{- dernierNoeud(Ch)}; \\ &\text{n1} <\text{- successeur(n)}; \\ &\text{while non(fin) et n1 est valide do} \\ &\text{if n1 est solution then} \\ &\text{Sol Ch Uf }[\text{n1}]; &\text{fin} <\text{- Vrai}; \\ &\text{else Liste } <\text{- Liste Uf }[\text{(Ch Uf }[\text{n1}])]; \\ &\text{n1} <\text{- successeur(n)}; \end{split}$$

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}

