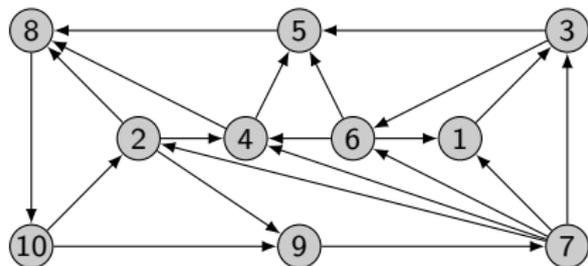


3.2 Problème de parcours

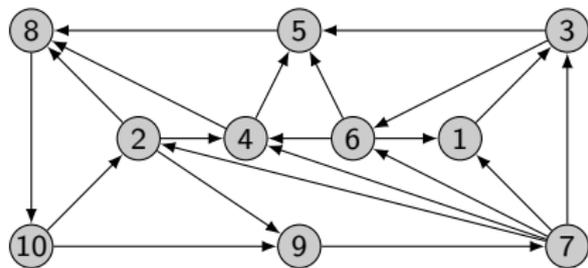
Parcours en largeur



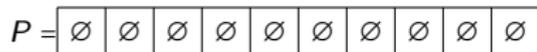
Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



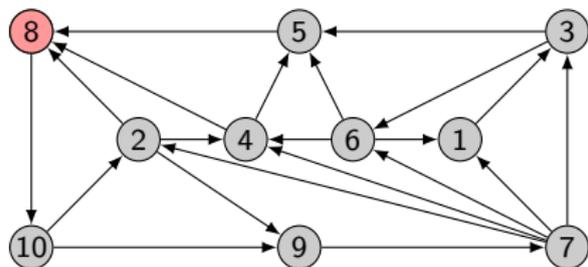
$L \Rightarrow () \rightarrow$



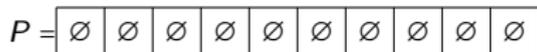
Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (8) \rightarrow$

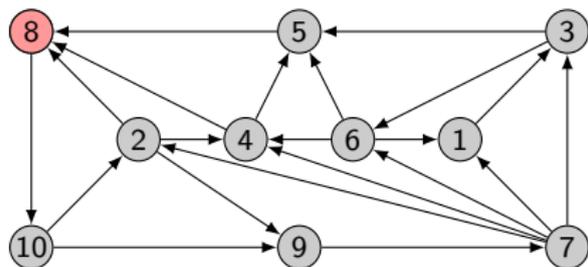


8

Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfileur  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



$L \Rightarrow () \rightarrow$

$x = 8$

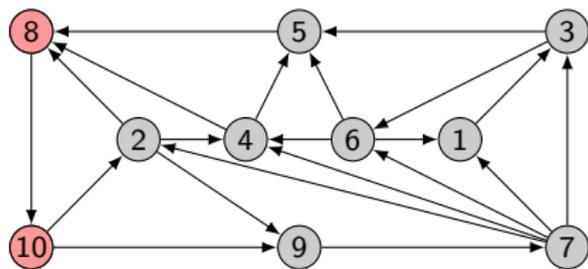
$P = [\emptyset \ \emptyset \ \emptyset]$

8

Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow () \rightarrow$

$x = 8$

$P = [\emptyset \ \emptyset \ \emptyset]$

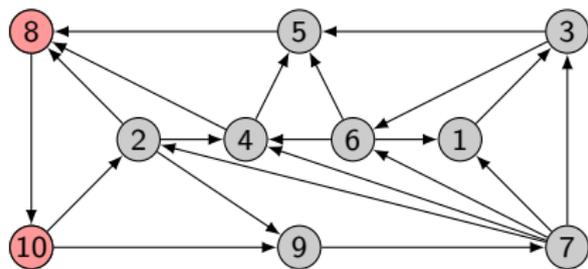
8

10

Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

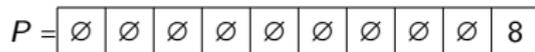
```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



$L \Rightarrow () \rightarrow$

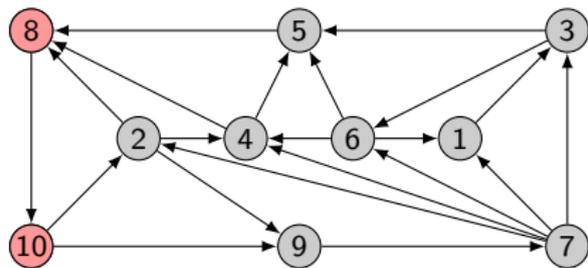
$x = 8$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

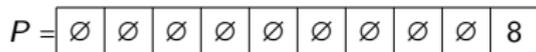
```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



$L \Rightarrow (10) \rightarrow$

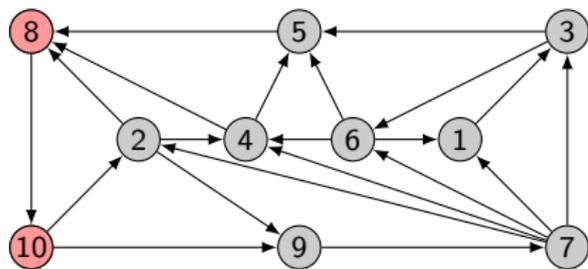
$x = 8$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



$L \Rightarrow () \rightarrow$

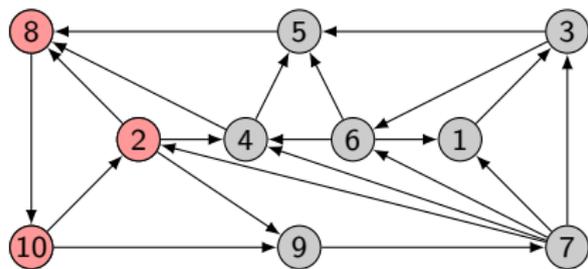
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

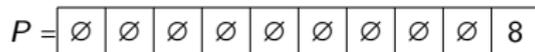
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow () \rightarrow$

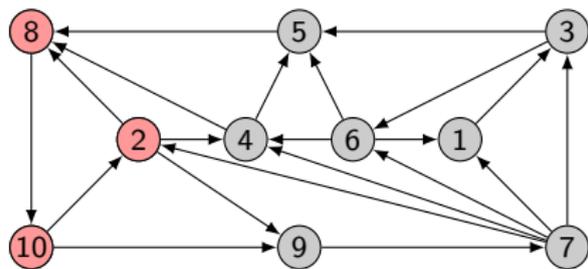
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

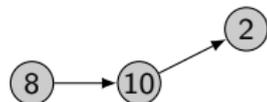
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow () \rightarrow$

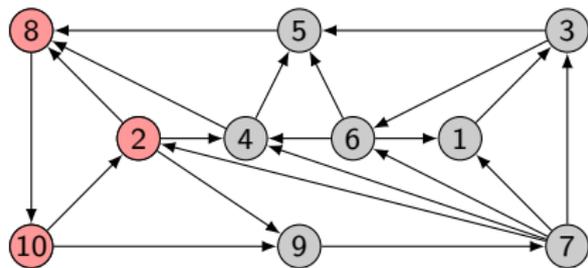
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

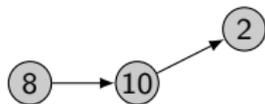
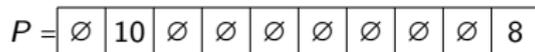
```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

Parcours en largeur



$L \Rightarrow (2) \rightarrow$

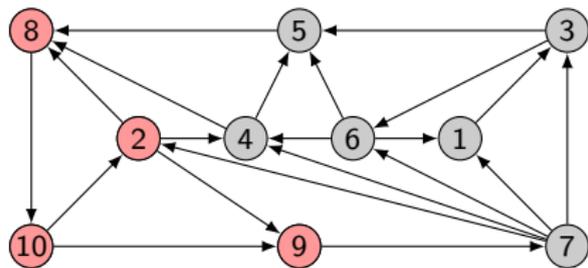
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

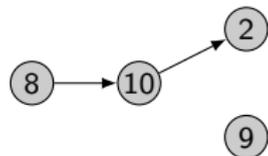
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (2) \rightarrow$

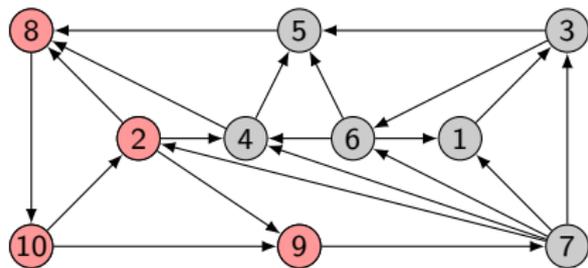
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

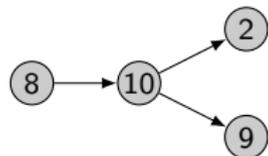
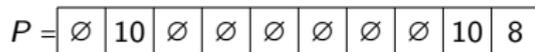
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (2) \rightarrow$

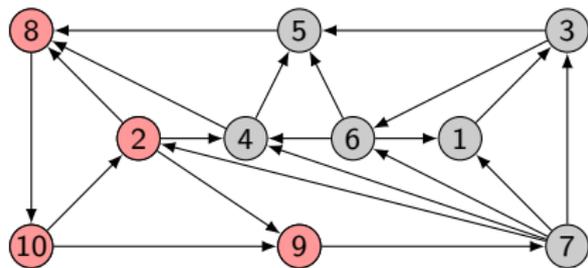
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

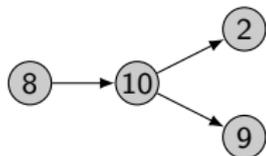
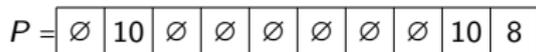
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (9, 2) \rightarrow$

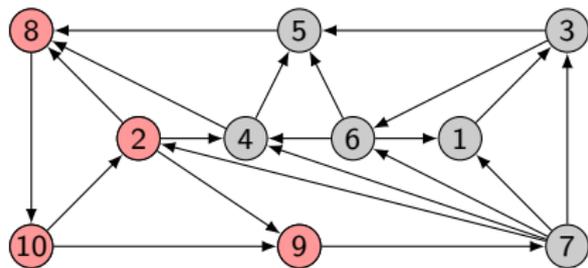
$x = 10$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

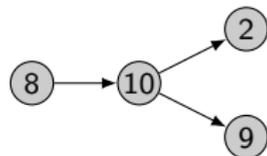
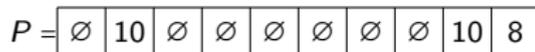
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (9) \rightarrow$

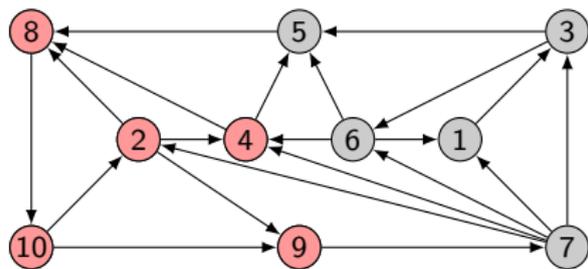
$x = 2$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

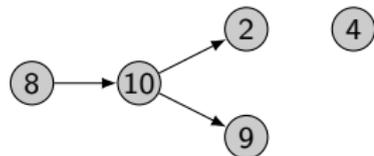
Parcours en largeur



$L \Rightarrow (9) \rightarrow$

$x = 2$

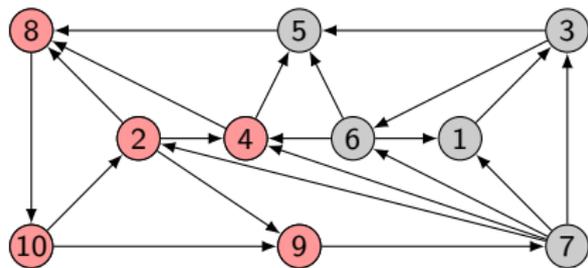
$P = [\emptyset \quad 10 \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

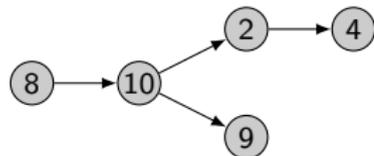
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (9) \rightarrow$

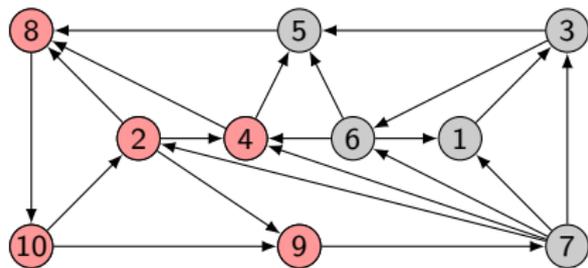
$x = 2$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

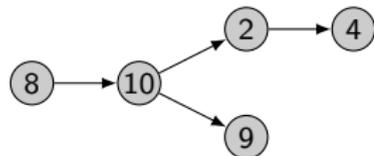
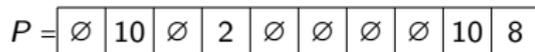
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (4, 9) \rightarrow$

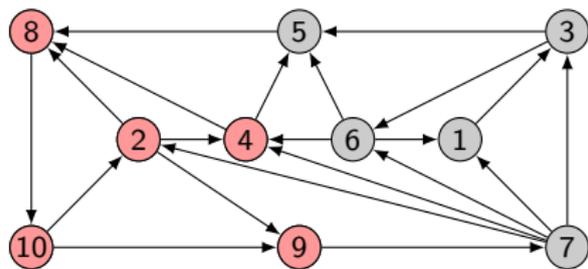
$x = 2$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

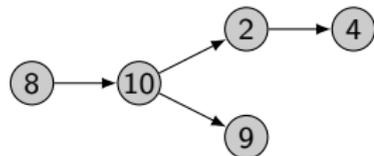
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (4) \rightarrow$

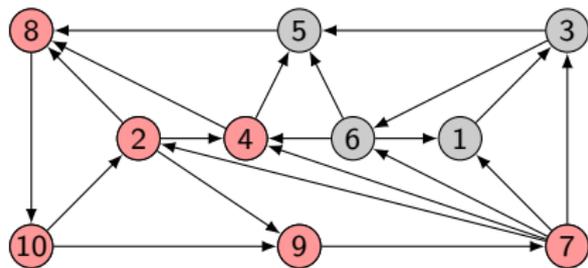
$x = 9$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

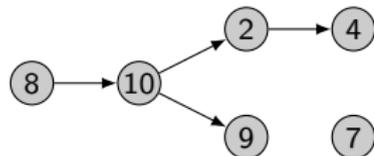
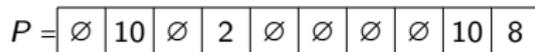
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (4) \rightarrow$

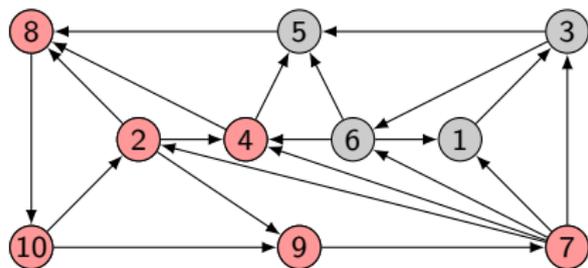
$x = 9$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

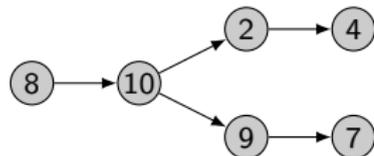
Parcours en largeur



$L \Rightarrow (4) \rightarrow$

$x = 9$

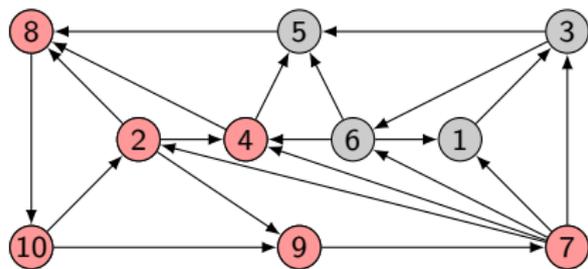
$P = [\emptyset, 10, \emptyset, 2, \emptyset, \emptyset, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

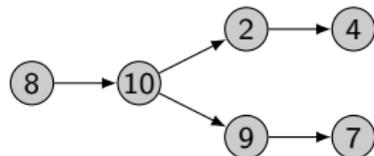
Parcours en largeur



$L \Rightarrow (7, 4) \rightarrow$

$x = 9$

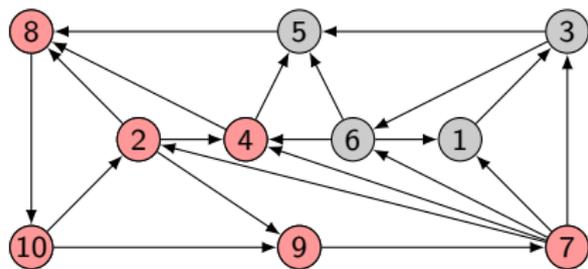
$P = [\emptyset \quad 10 \quad \emptyset \quad 2 \quad \emptyset \quad \emptyset \quad 9 \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

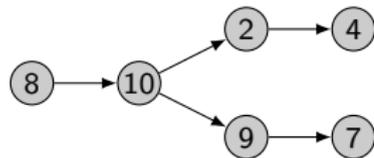
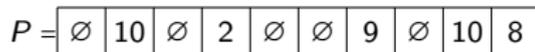
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (7) \rightarrow$

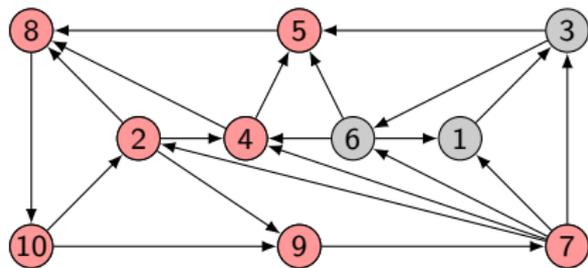
$x = 4$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

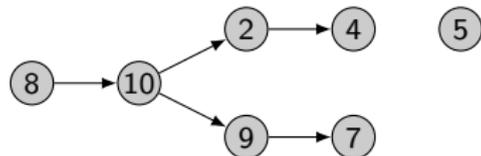
Parcours en largeur



$L \Rightarrow (7) \rightarrow$

$x = 4$

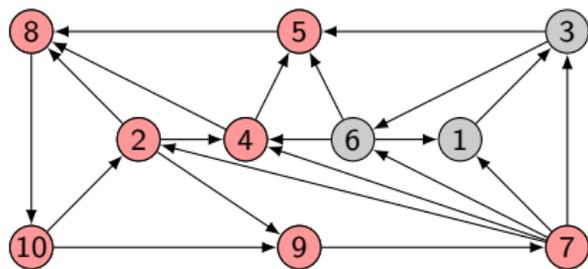
$P = [\emptyset \quad 10 \quad \emptyset \quad 2 \quad \emptyset \quad \emptyset \quad 9 \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

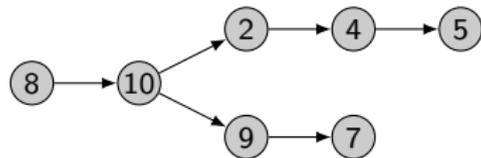
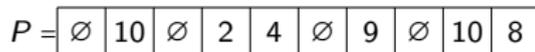
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (7) \rightarrow$

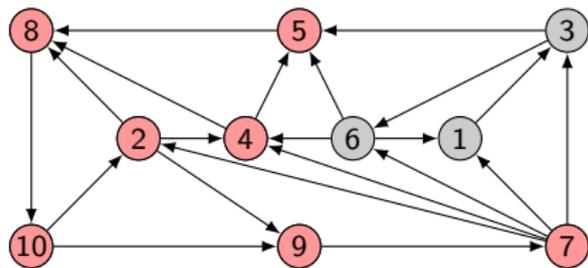
$x = 4$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

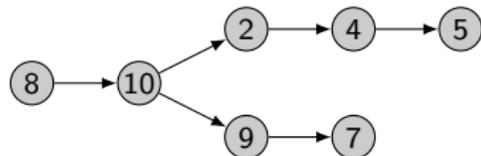
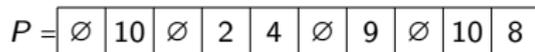
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (5, 7) \rightarrow$

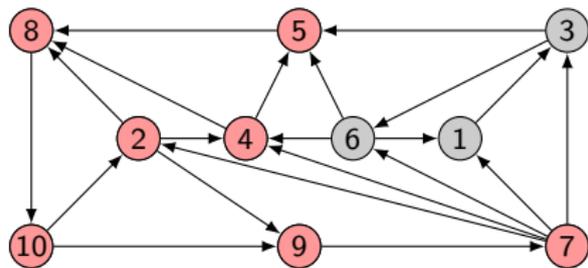
$x = 4$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

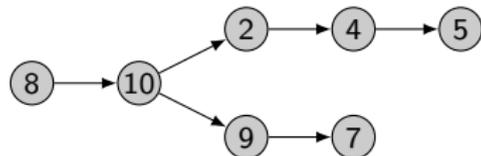
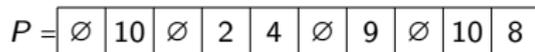
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (5) \rightarrow$

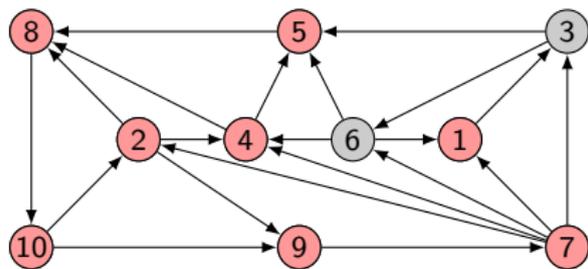
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

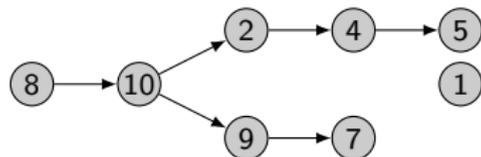
Parcours en largeur



$L \Rightarrow (5) \rightarrow$

$x = 7$

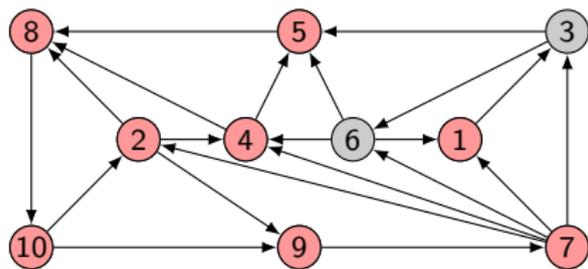
$P = [\emptyset, 10, \emptyset, 2, 4, \emptyset, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

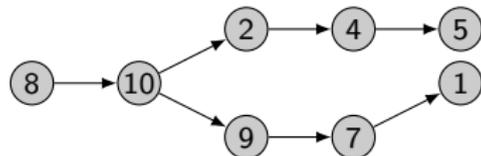
Parcours en largeur



$L \Rightarrow (5) \rightarrow$

$x = 7$

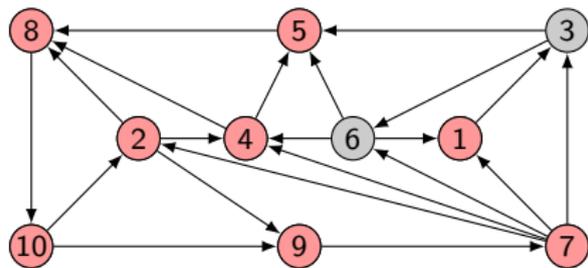
$P = [7 \quad 10 \quad \emptyset \quad 2 \quad 4 \quad \emptyset \quad 9 \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

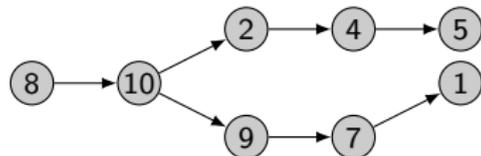
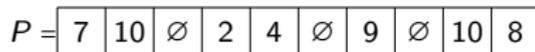
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (1, 5) \rightarrow$

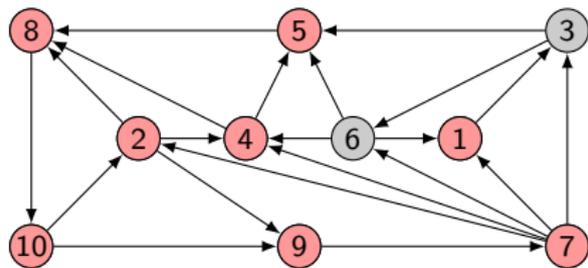
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

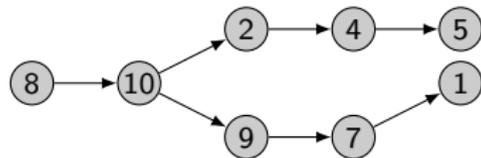
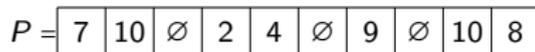
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (1, 5) \rightarrow$

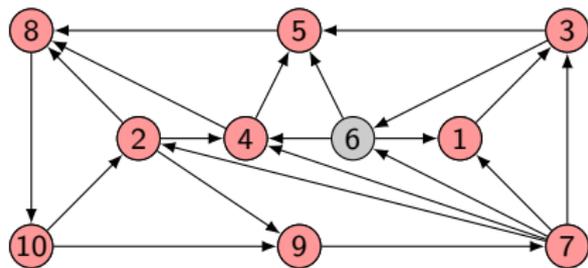
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

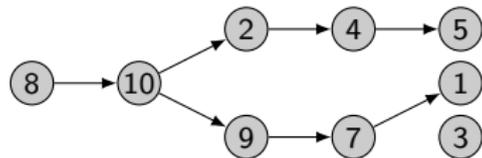
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (1, 5) \rightarrow$

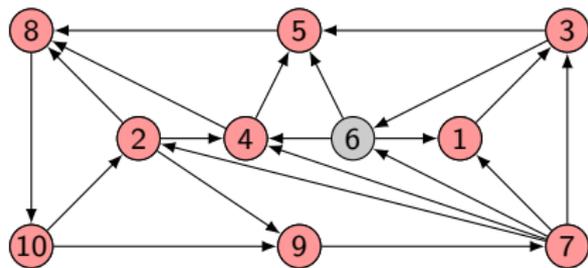
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

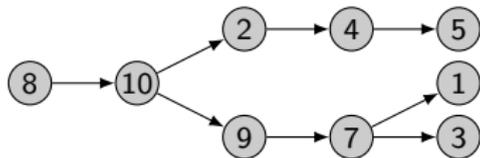
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (1, 5) \rightarrow$

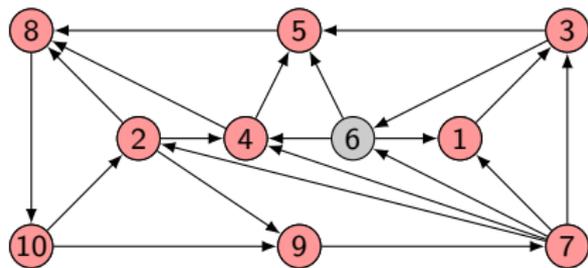
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

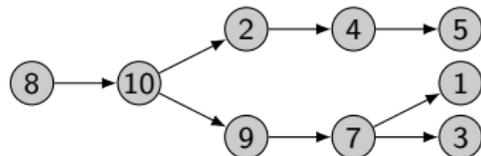
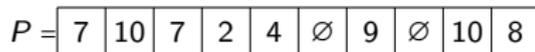
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (3, 1, 5) \rightarrow$

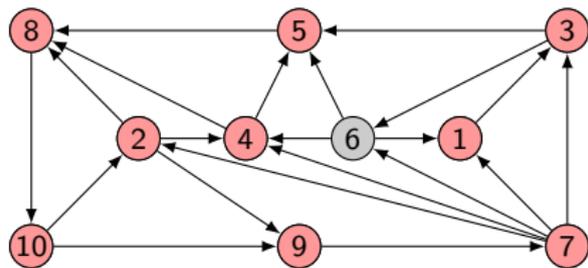
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

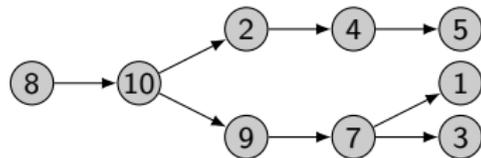
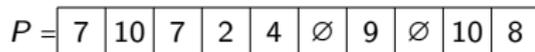
$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

Parcours en largeur



$L \Rightarrow (3, 1, 5) \rightarrow$

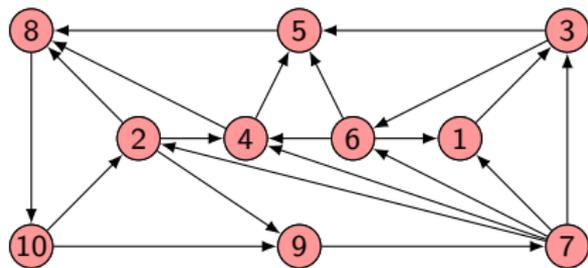
$x = 7$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

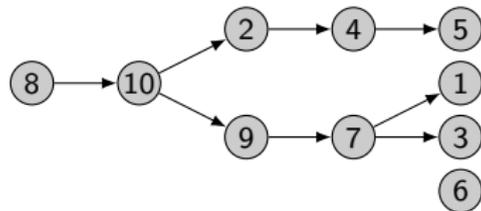
Parcours en largeur



$L \Rightarrow (3, 1, 5) \rightarrow$

$x = 7$

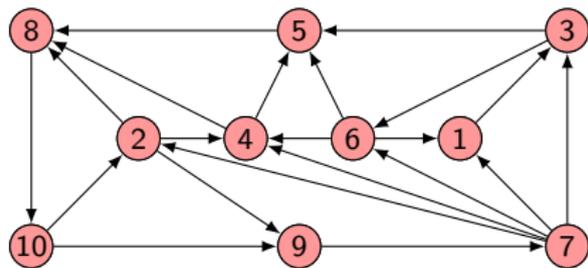
$P = [7 \quad 10 \quad 7 \quad 2 \quad 4 \quad \emptyset \quad 9 \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

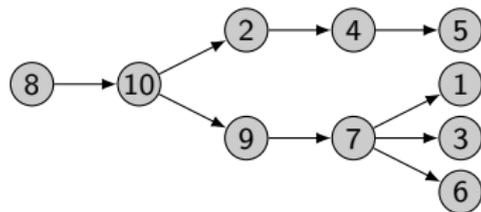
Parcours en largeur



$L \Rightarrow (3, 1, 5) \rightarrow$

$x = 7$

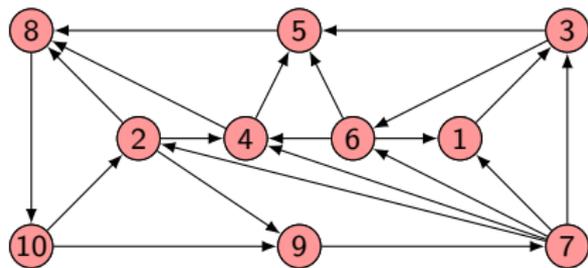
$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

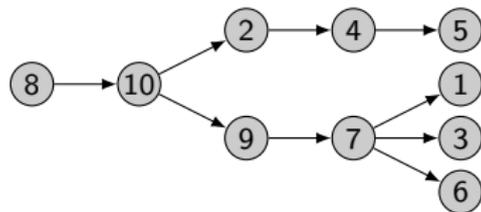
Parcours en largeur



$L \Rightarrow (6, 3, 1, 5) \rightarrow$

$x = 7$

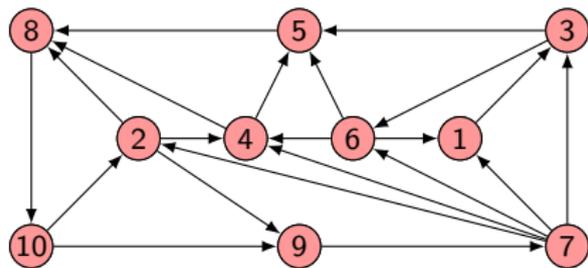
$P = [7 \quad 10 \quad 7 \quad 2 \quad 4 \quad 7 \quad 9 \quad \emptyset \quad 10 \quad 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

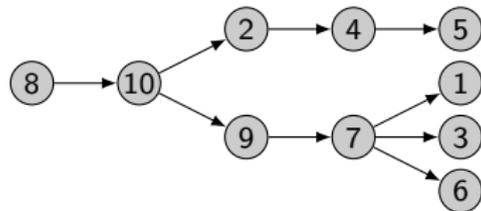
Parcours en largeur



$L \Rightarrow (6, 3, 1) \rightarrow$

$x = 5$

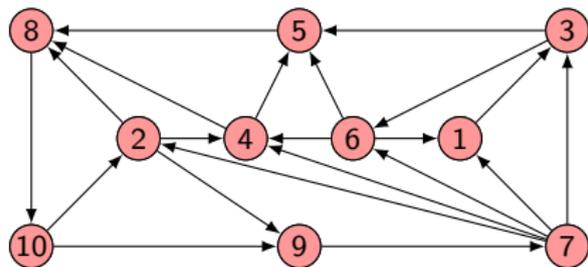
$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
L ← () File des sommets à traiter;  
P ← Tableau de taille |S| initialisé à ∅;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

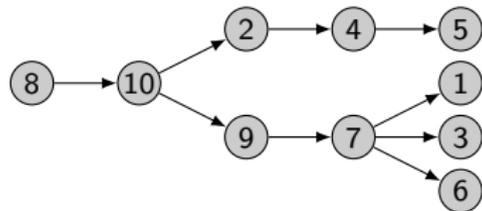
Parcours en largeur



$L \Rightarrow (6, 3) \rightarrow$

$x = 1$

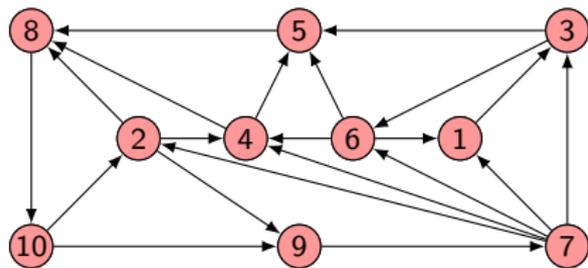
$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

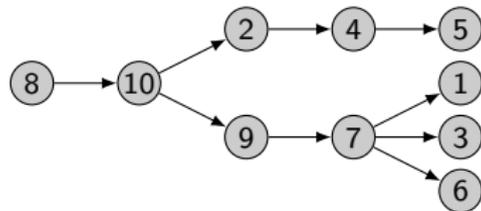
Parcours en largeur



$L \Rightarrow (6) \rightarrow$

$x = 3$

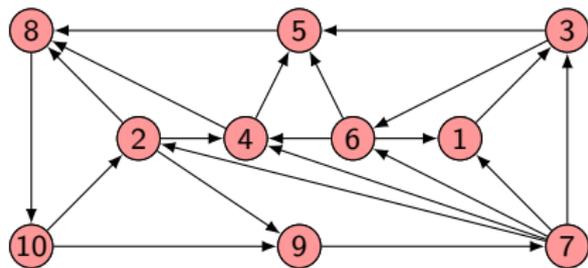
$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

```
 $L \leftarrow ()$  File des sommets à traiter;  
 $P \leftarrow$  Tableau de taille  $|S|$  initialisé à  $\emptyset$ ;  
Marquer le sommet  $s$  et l'enfiler dans  $L$ ;  
while  $L \neq \emptyset$  do  
   $x \leftarrow$  défiler le premier sommet de  $L$ ;  
  for  $y$  voisin non marqué de  $x$  do  
    Marquer  $y$ ;  
     $P(y) \leftarrow x$ ;  
    enfiler  $y$  dans  $L$ ;  
  end  
end  
Renvoyer  $P$ ;
```

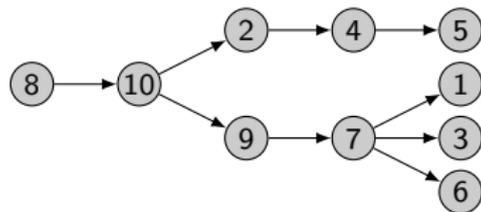
Parcours en largeur



$L \Rightarrow () \rightarrow$

$x = 6$

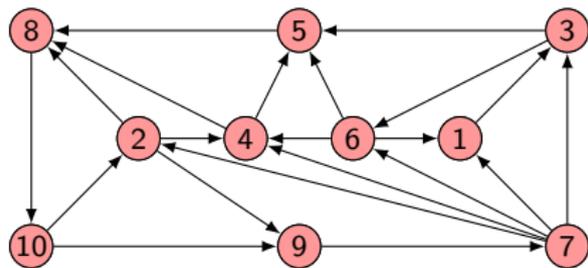
$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s
Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;
 $P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;
Marquer le sommet s et l'enfiler dans L ;
while $L \neq \emptyset$ **do**
 $x \leftarrow$ défiler le premier sommet de L ;
 for y voisin non marqué de x **do**
 Marquer y ;
 $P(y) \leftarrow x$;
 enfiler y dans L ;
 end
end
Renvoyer P ;

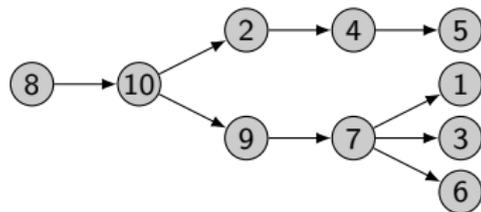
Parcours en largeur



$L \Rightarrow () \rightarrow$

$x = 6$

$P = [7, 10, 7, 2, 4, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s

Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ File des sommets à traiter;

$P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;

Marquer le sommet s et l'enfiler dans L ;

while $L \neq \emptyset$ **do**

$x \leftarrow$ défiler le premier sommet de L ;

for y voisin non marqué de x **do**

Marquer y ;

$P(y) \leftarrow x$;

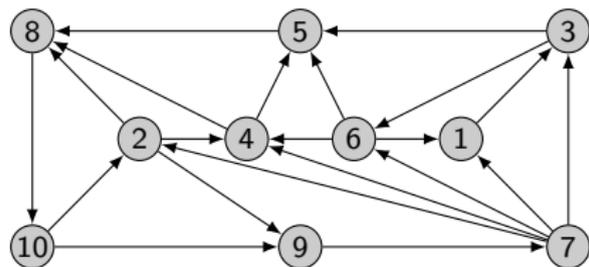
 enfiler y dans L ;

end

end

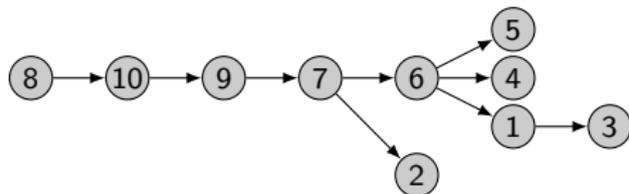
Renvoyer P ;

Parcours en profondeur



$L = \leftarrow ()$

$P = [6, 7, 1, 6, 6, 7, 9, \emptyset, 10, 8]$



Data: $G = (S, A)$ et un sommet s

Result: P : Liste des prédécesseurs dans l'arbre de parcours

$L \leftarrow ()$ Pile des sommets à traiter;

$P \leftarrow$ Tableau de taille $|S|$ initialisé à \emptyset ;

Enfiler s dans L ;

while $L \neq \emptyset$ **do**

$x \leftarrow$ dépiler le premier sommet de L ;

if x non marqué **then**

for y voisin de x et $P(y) \neq \emptyset$ **do**

$P(y) \leftarrow x$;

 Mettre y au début de L ;

end

 Marquer x ;

end

end

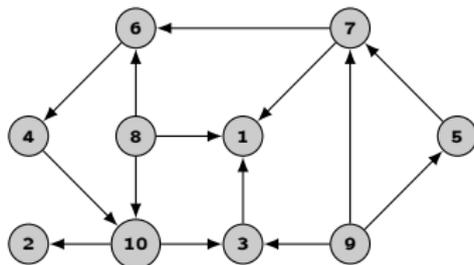
Renvoyer P ;

3.3 Initiation à la théorie des jeux

Un jeu combinatoire

Voici un jeu simple qui se joue sur à deux, sur un graphe orienté:

- on place un pion sur un sommet du graphe;
- à tour de rôle, chaque joueur doit déplacer le pion en suivant un arc du graphe;
- le premier joueur qui ne peut pas déplacer le pion a perdu.



On cherche à savoir s'il existe une stratégie gagnante pour l'un des joueurs, c'est à dire si il existe une méthode qui permet de le faire gagner quel que soit les coups réalisé par l'adversaire.

Jeux combinatoires

Tout jeu combinatoires à deux joueurs et à information complète peut être modélisé par le jeu précédent.

Jeux combinatoires

Tout jeu combinatoire à deux joueurs et à information complète peut être modélisé par le jeu précédent.

- Par *combinatoire* on entend de réflexion, c'est-à-dire que ce n'est pas un jeu d'habileté (type fléchettes) et sans hasard (ce qui exclut quasiment tous les jeux de cartes ou de dés).

Jeux combinatoires

Tout jeu combinatoire à deux joueurs et à information complète peut être modélisé par le jeu précédent.

- Par *combinatoire* on entend de réflexion, c'est-à-dire que ce n'est pas un jeu d'habileté (type fléchettes) et sans hasard (ce qui exclut quasiment tous les jeux de cartes ou de dés).
- A *deux joueurs* signifie que les deux joueurs jouent à tour de rôle (ce qui exclut des jeux type pierre-feuille-ciseau).

Jeux combinatoires

Tout jeu combinatoire à deux joueurs et à information complète peut être modélisé par le jeu précédent.

- Par *combinatoire* on entend de réflexion, c'est-à-dire que ce n'est pas un jeu d'habileté (type fléchettes) et sans hasard (ce qui exclut quasiment tous les jeux de cartes ou de dés).
- A *deux joueurs* signifie que les deux joueurs jouent à tour de rôle (ce qui exclut des jeux type pierre-feuille-ciseau).
- A *information complète* signifie que à tous moments les joueurs ont accès à l'état exact du jeu, il n'y a pas d'éléments cachés.

Jeux combinatoires

Tout jeux combinatoires à deux joueurs et à information complète peut être modélisé par le jeu précédent.

- Par *combinatoire* on entend de réflexion, c'est-à-dire que ce n'est pas un jeu d'habileté (type fléchettes) et sans hasard (ce qui exclut quasiment tous les jeux de cartes ou de dés).
- A *deux joueurs* signifie que les deux joueurs jouent à tour de rôle (ce qui exclut des jeux type pierre-feuille-ciseau).
- A *information complète* signifie que à tous moments les joueurs ont accès à l'état exact du jeu, il n'y a pas d'éléments cachés.

Proposition

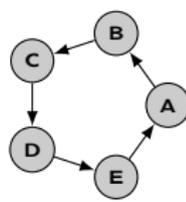
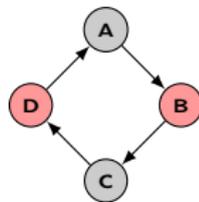
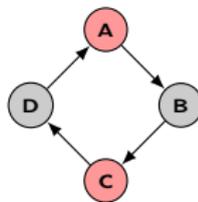
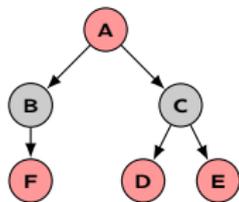
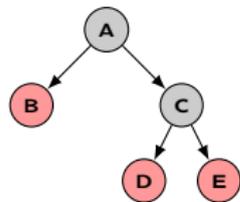
On se ramène au jeu précédent en considérant le graphe suivant:

- les sommets sont les états du systèmes;
- on place un arc entre deux sommets si on peut aller d'un état à l'autre.

Noyau d'un graphe orienté

Un ensemble de sommets $N \subset S$ est un *noyau* du graphe orienté $G = (S, A)$ s'il vérifie:

- pour tout $s \in N$ les successeurs de s ne sont pas dans N (N est *stable*),
- pour tout $s \in S \setminus N$ alors s admet un successeur dans N (N est *absorbant*).



Théorème

Tout joueur dont la position initiale n'est pas dans le noyau a une stratégie non perdante.

Noyau d'un graphe orienté

Théorème

Un graphe orienté sans circuit possède un unique noyau.

Noyau d'un graphe orienté

Théorème

Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

Noyau d'un graphe orienté

Théorème

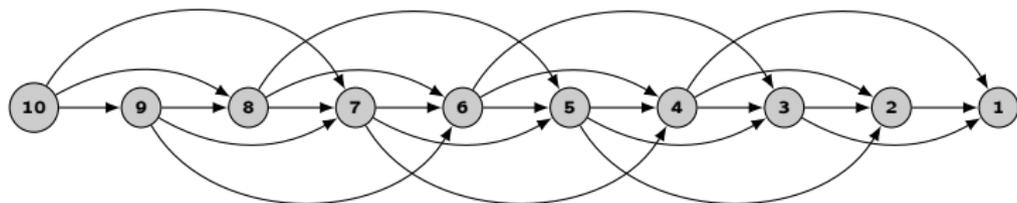
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

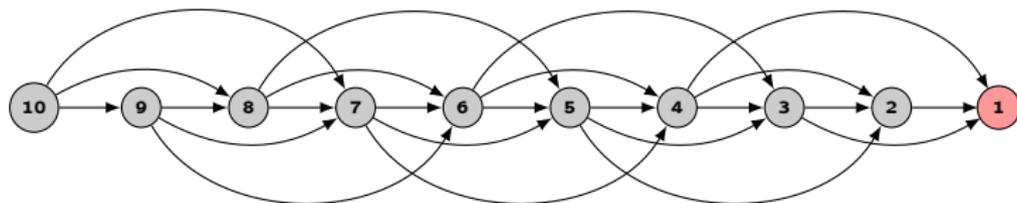
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

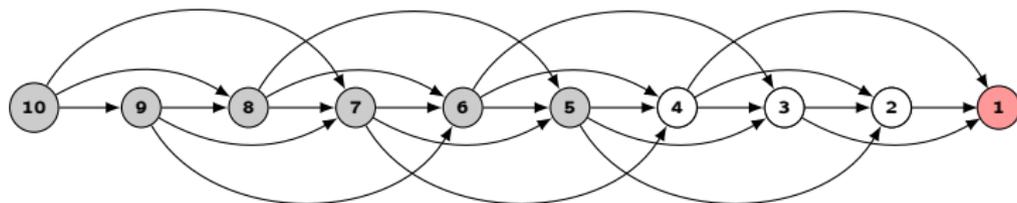
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

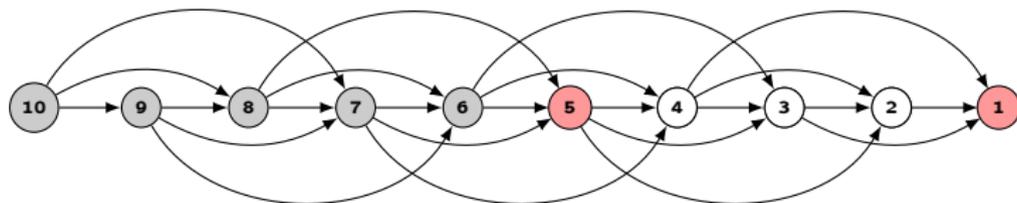
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

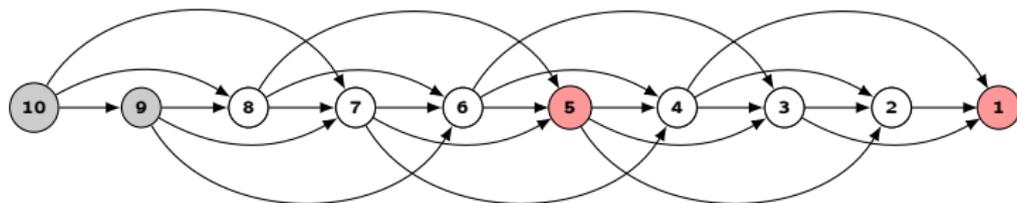
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

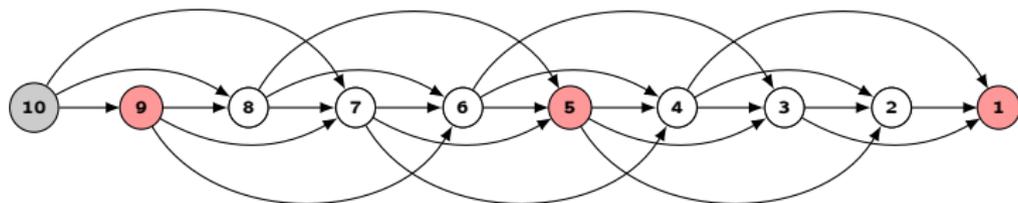
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

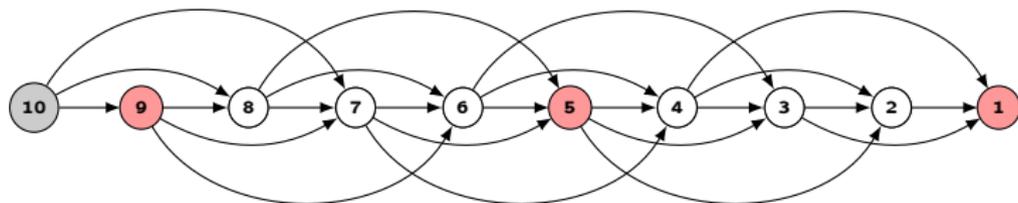
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Noyau d'un graphe orienté

Théorème

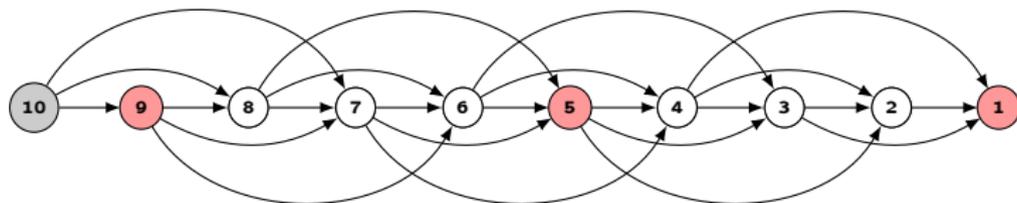
Un graphe orienté sans circuit possède un unique noyau.

Preuve: On remarque:

- un graphe sans circuit possède au moins un puits,
- un noyau contient nécessairement tous les puits,
- les prédécesseurs des puits ne peuvent pas être dans un noyau.

On construit le noyau N ainsi :

- on met tous les puits dans N ,
- on supprime de G tous les puits et tous leurs prédécesseurs,
- on recommence avec le nouveau graphe obtenu (qui est toujours sans circuit).



Jeu de Nim



Règles:

- 20 batons initialement
- à chaque coup un joueur prend 1,2 ou 3 batons
- le dernier qui prend un baton a perdu.

Taux de réussite à "Fort Boyaux":

39% sur 79 parties

Jeu de Nim



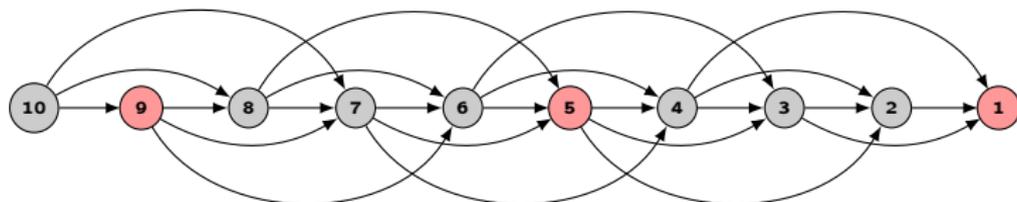
Règles:

- 20 batons initialement
- à chaque coup un joueur prend 1,2 ou 3 batons
- le dernier qui prend un baton a perdu.

Taux de réussite à "Fort Boyaux":

39% sur 79 parties

Cas de 10 bâtons:



Jeu de Nim



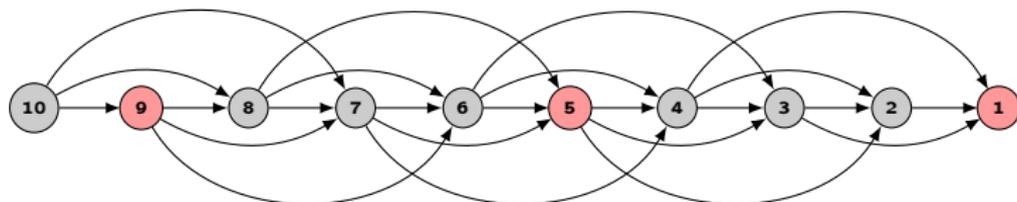
Règles:

- 20 batons initialement
- à chaque coup un joueur prend 1,2 ou 3 batons
- le dernier qui prend un baton a perdu.

Taux de réussite à "Fort Boyaux":

39% sur 79 parties

Cas de 10 bâtons:



Cas de n bâtons:

On montre par récurrence que $N = \{4k + 1 : k \in \mathbb{N}\}$.

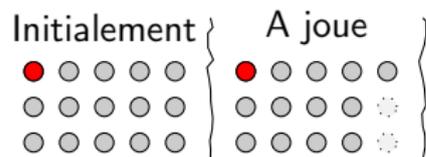
Ainsi le joueur 2 a une stratégie gagnante si et seulement si $n - 1$ divise 4.

Jeu de Chomp

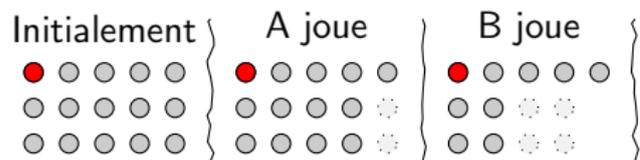
Initialement



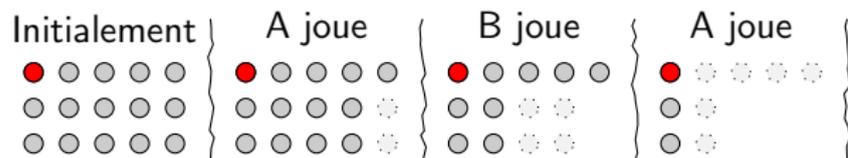
Jeu de Chomp



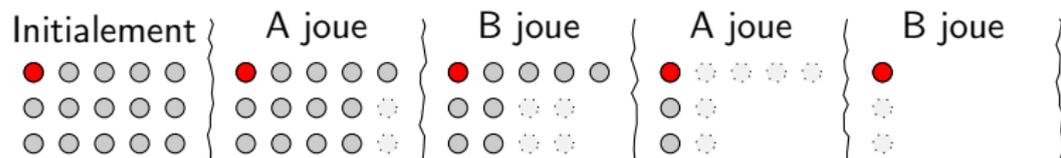
Jeu de Chomp



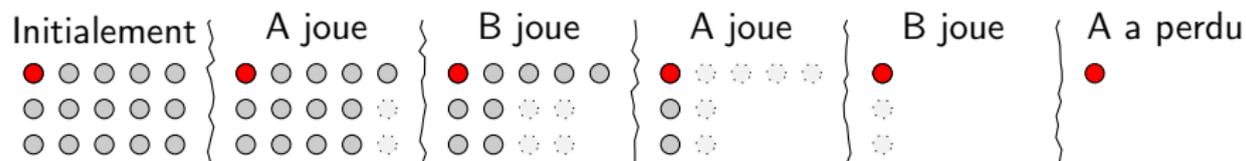
Jeu de Chomp



Jeu de Chomp



Jeu de Chomp



Quel joueur a une stratégie gagnante?

Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie

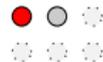
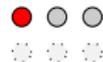
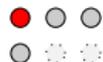
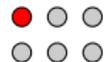


Stratégie: au premier coup, le joueur 1 mange tout les carrés mangeable!

Jeu de Chomp

Quel joueur a une stratégie gagnante?

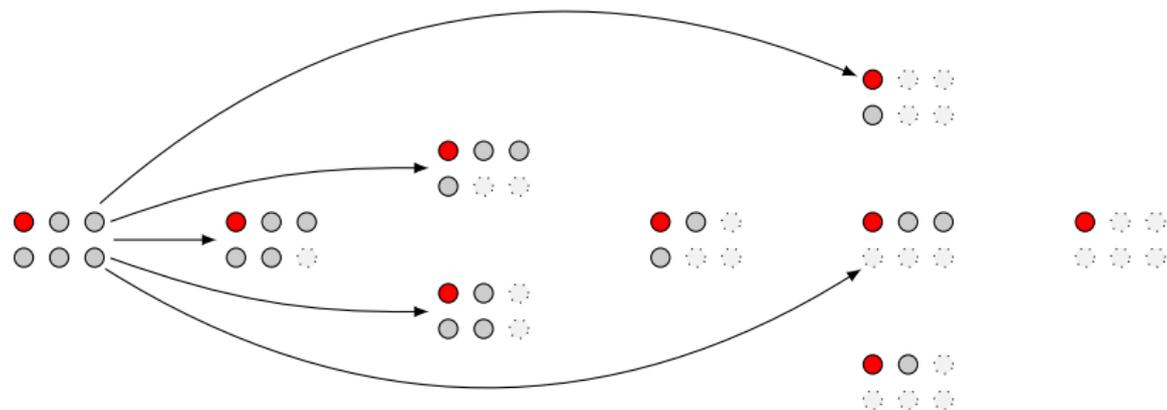
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

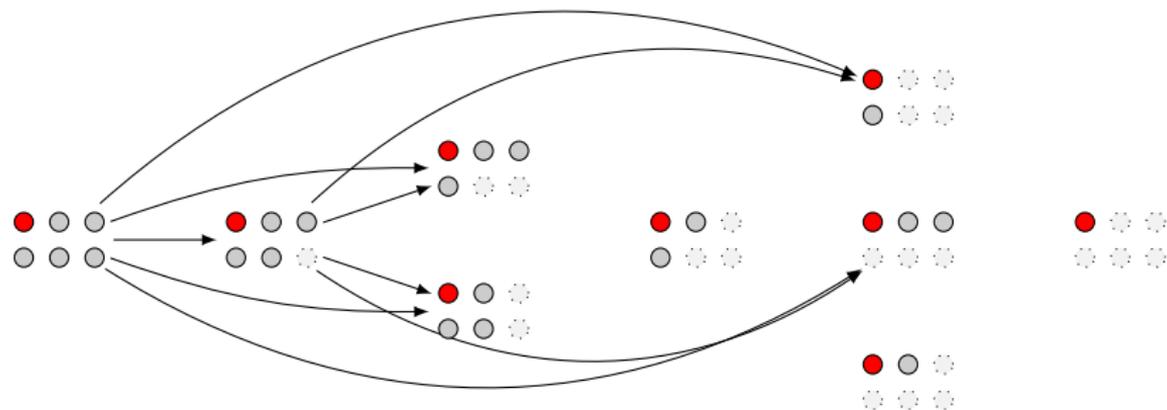
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

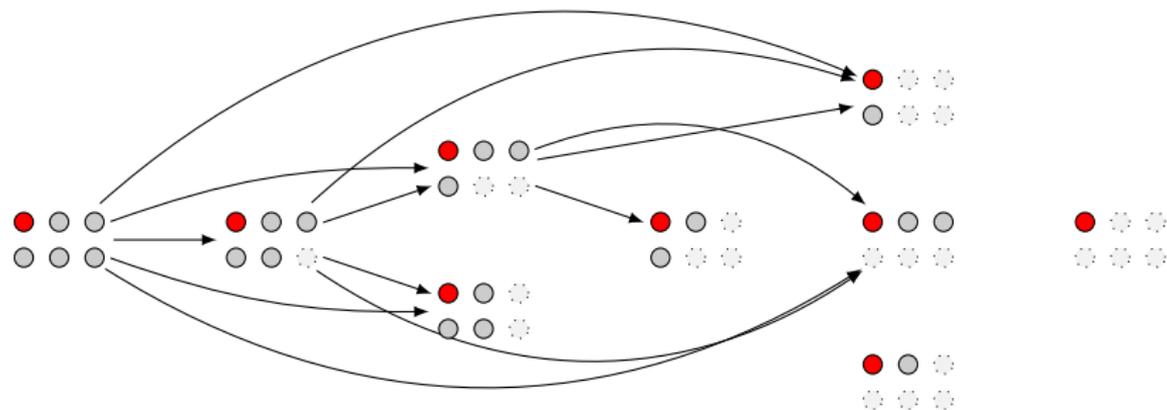
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

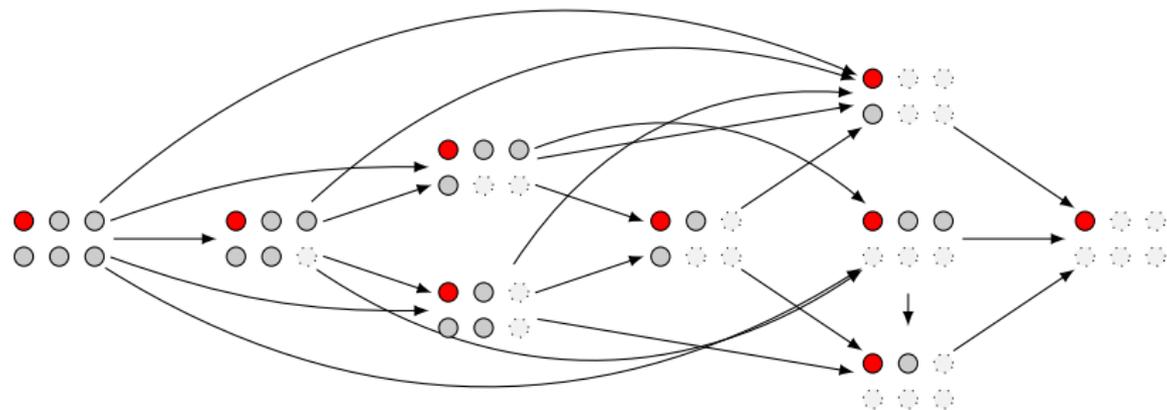
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

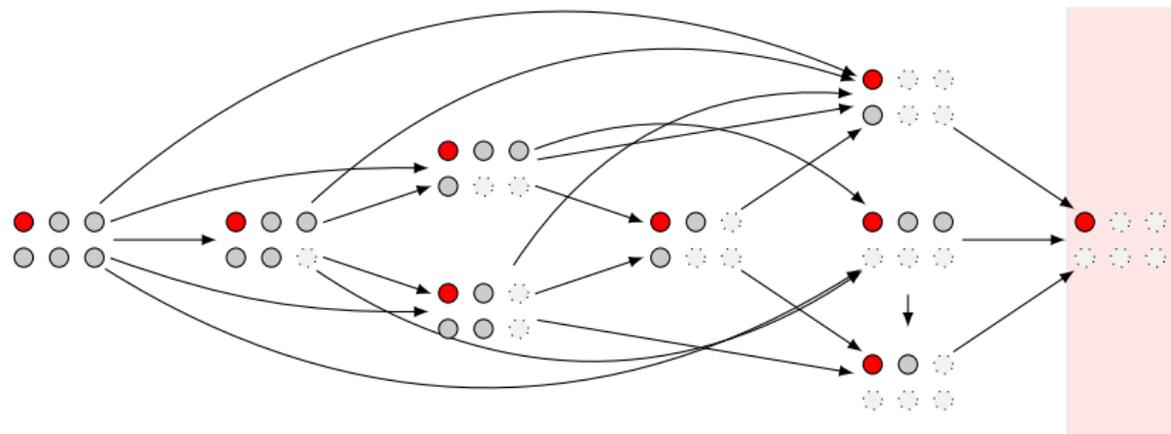
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

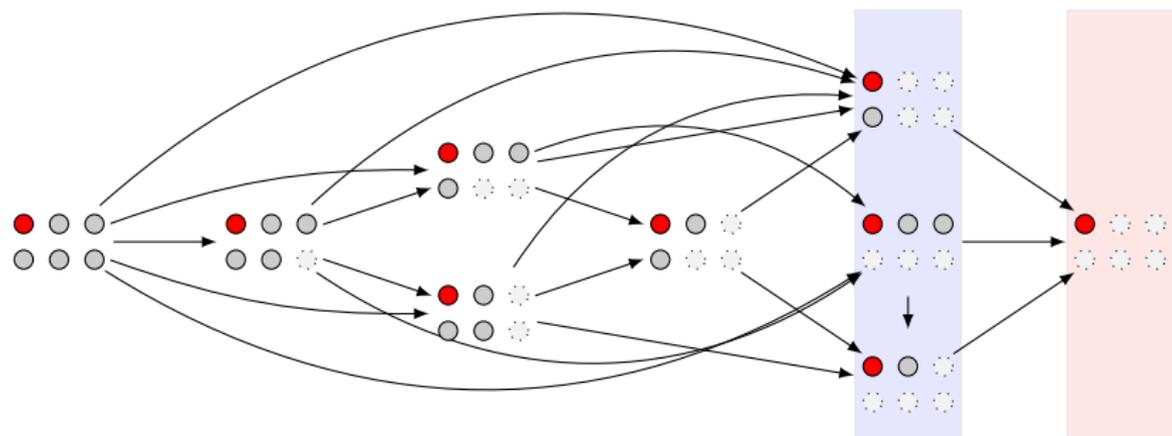
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

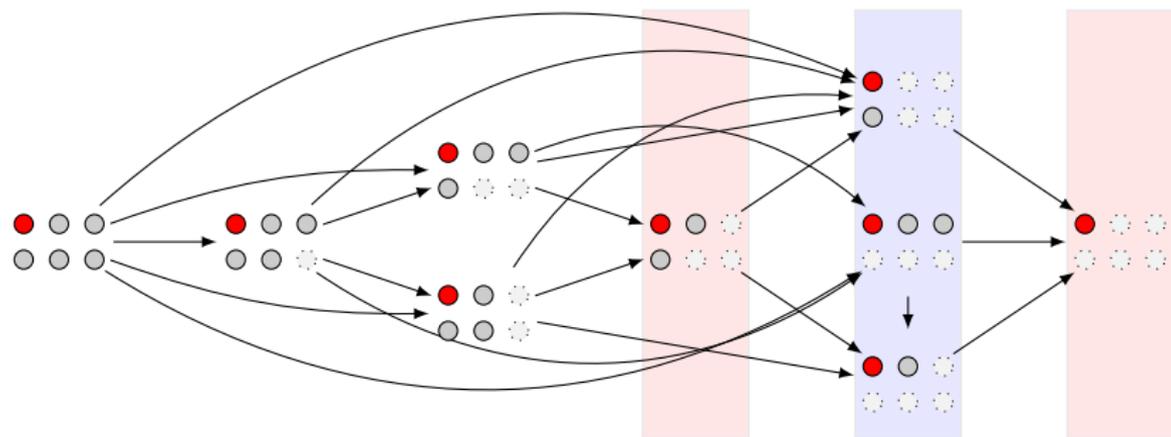
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

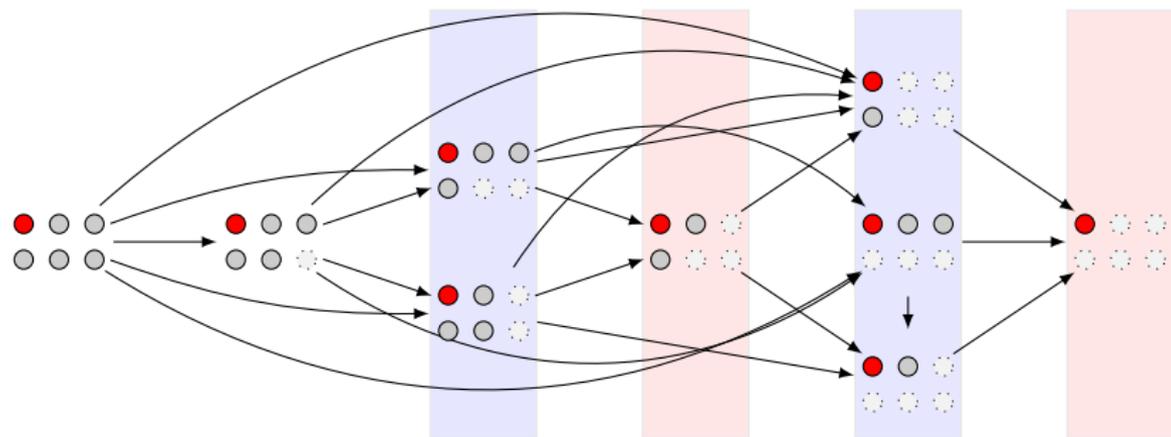
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

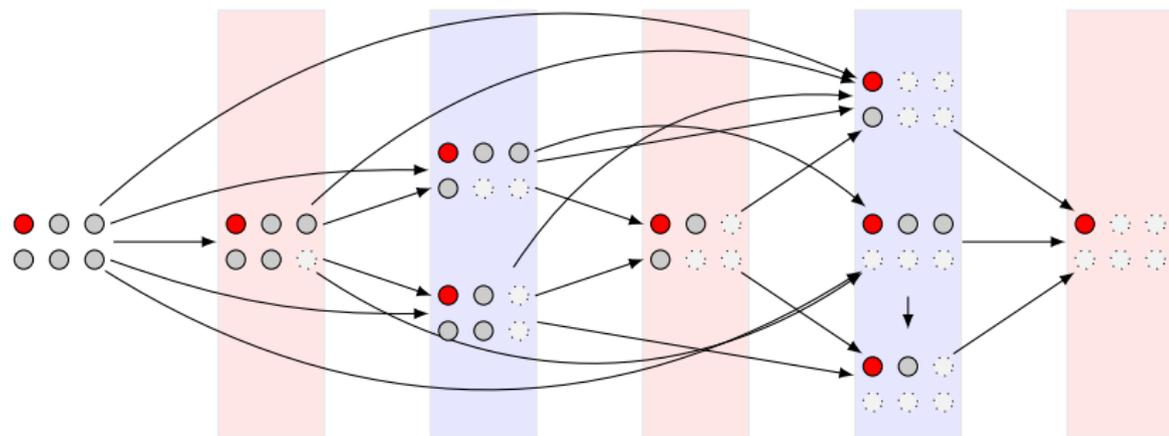
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

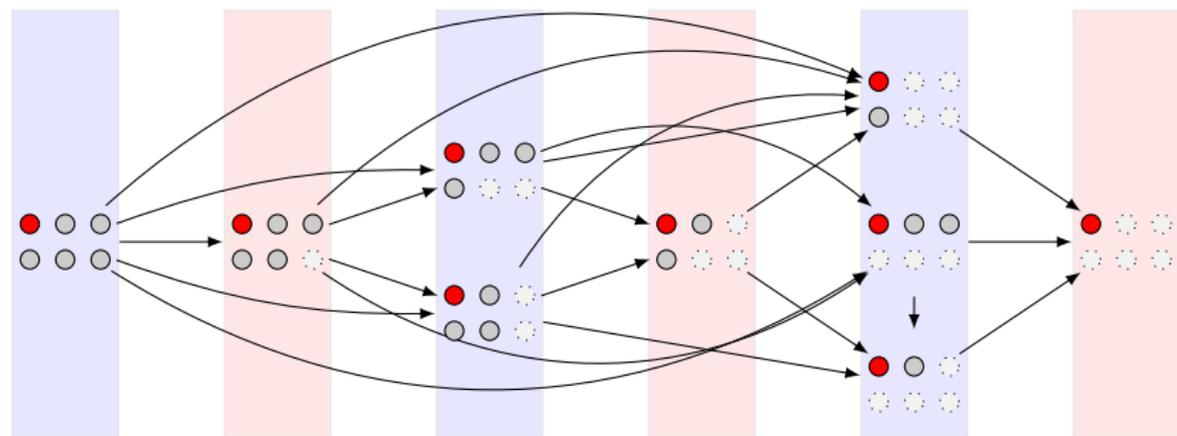
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

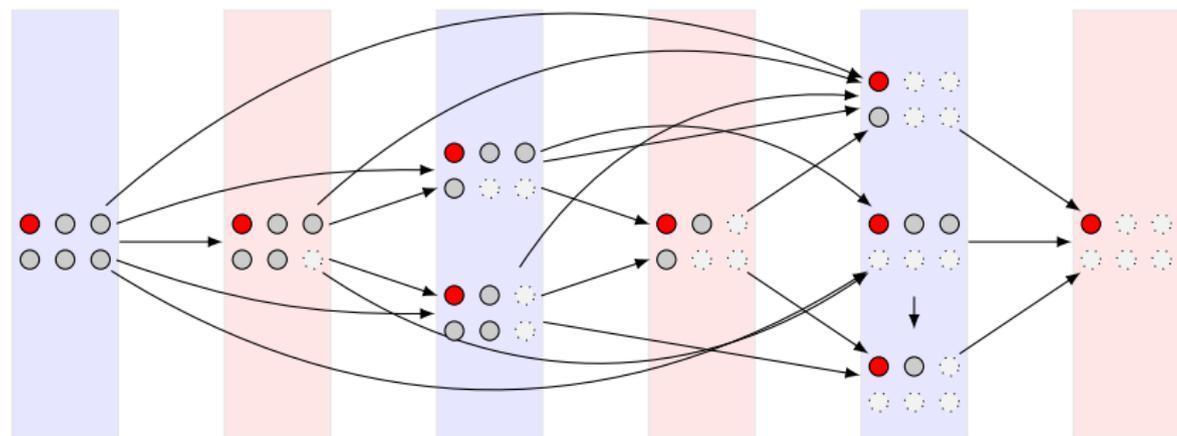
- Tablette $n \times 1$: Le joueur 1 a une stratégie gagnante
- Tablette 3×2



Jeu de Chomp

Quel joueur a une stratégie gagnante?

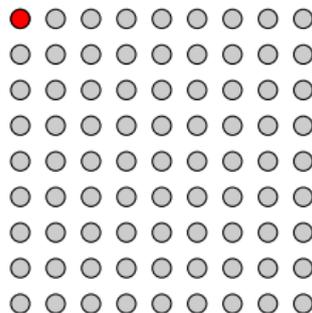
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie



Jeu de Chomp

Quel joueur a une stratégie gagnante?

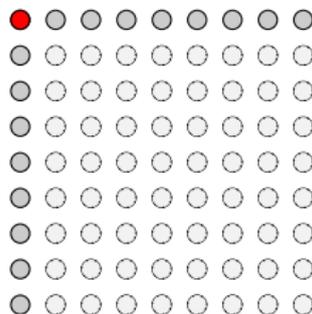
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

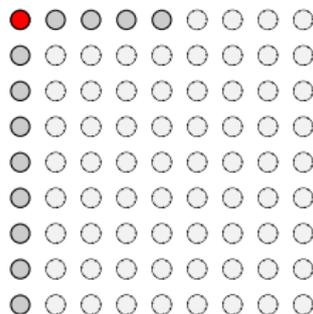
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

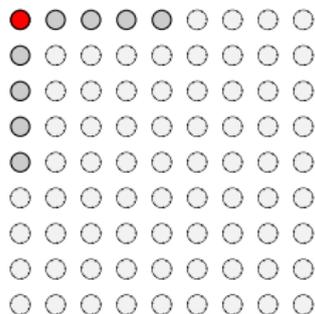
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

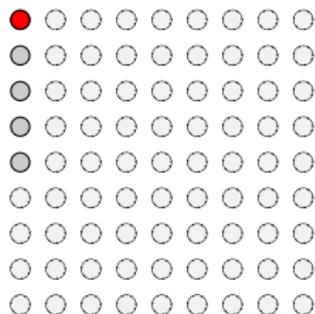
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

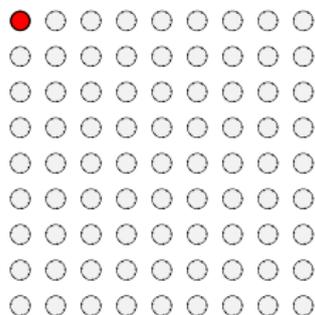
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$



Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie



Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$

Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie

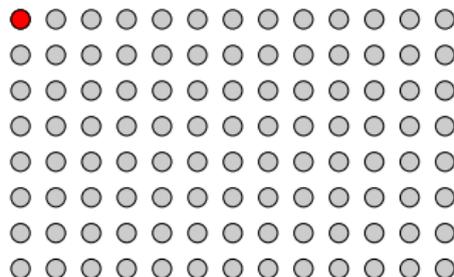
Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie

Preuve (par l'absurde):

Supposons que le joueur 2 a une stratégie gagnante.



Jeu de Chomp

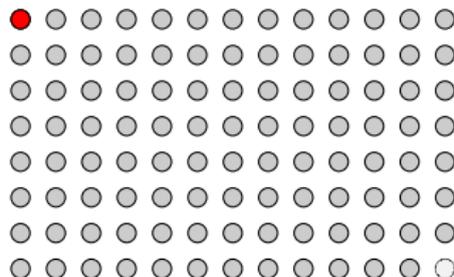
Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie

Preuve (par l'absurde):

Supposons que le joueur 2 a une stratégie gagnante.

Le joueur 1 mange le carré en bas à droite.



Jeu de Chomp

Quel joueur a une stratégie gagnante?

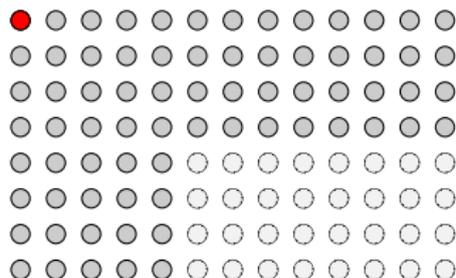
- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie

Preuve (par l'absurde):

Supposons que le joueur 2 a une stratégie gagnante.

Le joueur 1 mange le carré en bas à droite.

Le joueur 2 joue un coup pour amener le joueur 1 dans le noyau.



Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie

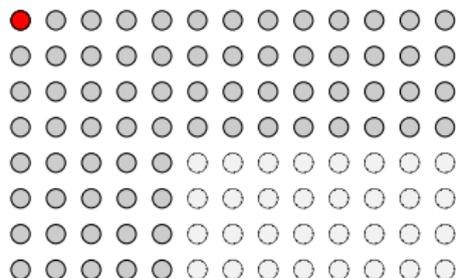
Preuve (par l'absurde):

Supposons que le joueur 2 a une stratégie gagnante.

Le joueur 1 mange le carré en bas à droite.

Le joueur 2 joue un coup pour amener le joueur 1 dans le noyau.

Mais le jouer 1 aurait pu jouer ce coup au départ! (on parle de vol de stratégie)



Jeu de Chomp

Quel joueur a une stratégie gagnante?

- Tablette $n \times 1$: Le joueur 1 a une stratégie
- Tablette 3×2 : Le joueur 1 a une stratégie
- Tablette $n \times n$: Le joueur 1 a une stratégie
- Tablette $m \times n$: Le joueur 1 a une stratégie
- Tablette $n \times 2$, $n \times 3$: essayer de trouver la **méthode** pour gagner à tout les coups!