Recherche Optimale de Trou Noir avec Cailloux

Paola Flocchini[1] David Ilcinkas[2] Nicola Santoro[3]

[1]SITE, University of Ottawa, Canada

[2]CNRS, LaBRI, Université Bordeaux I, France

[3]School of Computer Science, Carleton University, Canada

AlgoTel '08, 16 mai 2008

Problème

Trou noir dans un réseau

Nœud capturant et éliminant tout agent mobile y entrant.
Motivations :

- Site détruit et/ou dangereux pour un robot physique
- Nœud infecté par un virus “destructeur”
- Nœud crashé ou silencieux

Recherche de trou noir

A partir d'un nœud sain (base de départ), une équipe d'agents mobiles doit trouver l'emplacement du/des trous noirs.
Performance : nb d'agents, nb de déplacements, temps

Problème

Trou noir dans un réseau

Nœud capturant et éliminant tout agent mobile y entrant.
Motivations :

- Site détruit et/ou dangereux pour un robot physique
- Nœud infecté par un virus "destructeur"
- Nœud crashé ou silencieux

Recherche de trou noir

A partir d'un nœud sain (base de départ), une équipe d'agents mobiles doit trouver l'emplacement du/des trous noirs.

Performance : nb d'agents, nb de déplacements, temps

Problème

Trou noir dans un réseau

Nœud capturant et éliminant tout agent mobile y entrant.
Motivations :

- Site détruit et/ou dangereux pour un robot physique
- Nœud infecté par un virus "destructeur"
- Nœud crashé ou silencieux

Recherche de trou noir

A partir d'un nœud sain (base de départ), une équipe d'agents mobiles doit trouver l'emplacement du/des trous noirs.
Performance : nb d'agents, nb de déplacements, temps

(A)synchrone

Synchrone

Les agents agissent tous les tops d'horloge.

$\Longrightarrow$ Possibilité d'attendre un agent parti explorer une arête potentiellement dangereuse.

Asynchrone

Toute action prend un temps fini mais non borné.

$\Longrightarrow$ Impossible *a priori* de distinguer un lien très lent d'un lien menant au trou noir.

$\Longrightarrow$ Inutile d'attendre un agent.

$\Longrightarrow$ Pb équivalent à l'exploration de réseaux dangereux.

(A)synchrone

Synchrone

Les agents agissent tous les tops d'horloge.

$\Longrightarrow$ Possibilité d'attendre un agent parti explorer une arête potentiellement dangereuse.

Asynchrone

Toute action prend un temps fini mais non borné.

$\Longrightarrow$ Impossible *a priori* de distinguer un lien très lent d'un lien menant au trou noir.

$\Longrightarrow$ Inutile d'attendre un agent.

$\Longrightarrow$ Pb équivalent à l'exploration de réseaux dangereux.

Hypothèses nécessaires (en asynchrone)

Connaissance nécessaire

- Connaissance du nombre de trous noirs
  $\longrightarrow$ Ici, exactement un trou noir
- Connaissance du nombre de sommets n

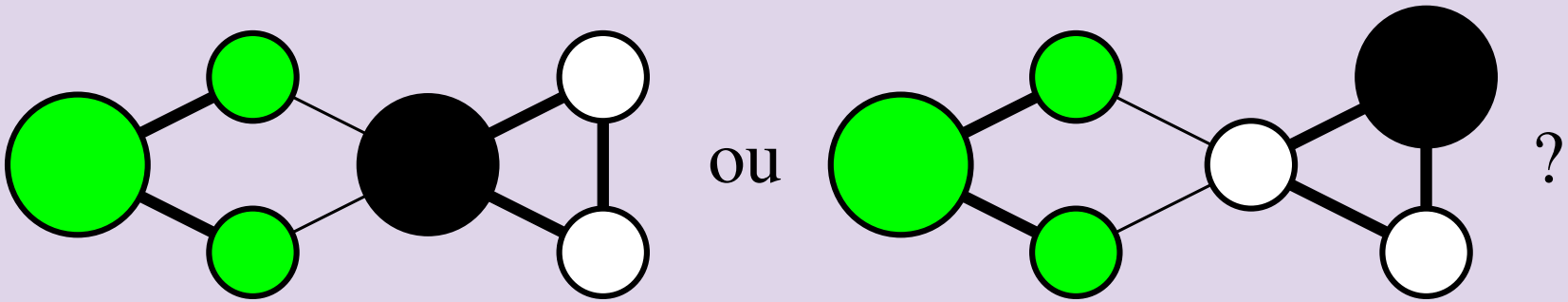
2-sommet-connexe

Hypothèses nécessaires (en asynchrone)

Connaissance nécessaire

- Connaissance du nombre de trous noirs
  $\longrightarrow$ Ici, exactement un trou noir
- Connaissance du nombre de sommets n

2-sommet-connexe



Modèle (toujours en asynchrone)

Communication/coordination

Pas de communication directe mais

Tableaux blancs (zones de mémoire sur chaque nœud).

ou Marqueurs d'arêtes (marquent un port spécifique).

ou Cailloux (marqueurs de sommets).

Pas de calcul

- Observation : présence d'un caillou/message sur le nœud.
- Interaction : dépot ou retrait d'un caillou/message.
- Action : déplacement le long d'une arête (ou terminaison).

Cet ensemble d'étapes (pas de calcul) est exécuté de façon atomique en exclusion mutuelle.

Modèle (toujours en asynchrone)

Communication/coordination

Pas de communication directe mais

Tableaux blancs (zones de mémoire sur chaque nœud).

ou Marqueurs d'arêtes (marquent un port spécifique).

ou Cailloux (marqueurs de sommets).

Pas de calcul

- Observation : présence d'un caillou/message sur le nœud.
- Interaction : dépot ou retrait d'un caillou/message.
- Action : déplacement le long d'une arête (ou terminaison).

Cet ensemble d'étapes (pas de calcul) est exécuté de façon atomique en exclusion mutuelle.

Résultats précédents (asynchrone, tableaux blancs)

Résultats de base

[Dobrev, Flocchini, Prencipe, Santoro. Dist. Comp. 2006]

	nb d'agents	nb de déplacements
Pas d'info (à part n)	$\Delta + 1$	$\Theta(n^2)$
Sens d'orientation	2	$\Theta(n^2)$
Information complète	2	$\Theta(n \log n)$

Différentes améliorations (avec information complète)

[Dobrev, Flocchini, Santoro. SIROCCO 2004]

- Topologies spécifiques (mais pas l'anneau)
- Nb de déplacements : $O(n)$

[Dobrev et al. Networks 2006]

- Nb de déplacements : $O(n + D \log D)$ (D = diamètre)

Résultats précédents (asynchrone, tableaux blancs)

Résultats de base

[Dobrev, Flocchini, Prencipe, Santoro. Dist. Comp. 2006]

	nb d'agents	nb de déplacements
Pas d'info (à part n)	$\Delta + 1$	$\Theta(n^2)$
Sens d'orientation	2	$\Theta(n^2)$
Information complète	2	$\Theta(n \log n)$

Différentes améliorations (avec information complète)

[Dobrev, Flocchini, Santoro. SIROCCO 2004]

- Topologies spécifiques (mais pas l'anneau)
- Nb de déplacements : $O(n)$

[Dobrev et al. Networks 2006]

- Nb de déplacements : $O(n + D \log D)$ (D = diamètre)

Résultats précédents (asynchrone)

Conditions minimales sur la coordination des agents.

Marqueurs d'arêtes, anneaux, liens FIFO

[Dobrev, Santoro, Shi. CIAC 2006]

- $O(1)$ marqueurs d'arêtes en tout, jusqu'à 3 par nœud
- Nb de déplacements : $\Theta(n \log n)$

Marqueurs d'arêtes, graphes arbitraires, liens FIFO

[Dobrev, Flocchini, Kralovic, Santoro. IFIP TCS 2006]

- Aucune information (à part n)
- Nb d'agents : $\Delta + 1$
- Nb de déplacements : env. $O(n^{10})$

Pb : La coordination à l'aide de cailloux est-elle suffisante ?

Résultats précédents (asynchrone)

Conditions minimales sur la coordination des agents.

Marqueurs d'arêtes, anneaux, liens FIFO

[Dobrev, Santoro, Shi. CIAC 2006]

- $O(1)$ marqueurs d'arêtes en tout, jusqu'à 3 par nœud
- Nb de déplacements : $\Theta(n \log n)$

Marqueurs d'arêtes, graphes arbitraires, liens FIFO

[Dobrev, Flocchini, Kralovic, Santoro. IFIP TCS 2006]

- Aucune information (à part n)
- Nb d'agents : $\Delta + 1$
- Nb de déplacements : env. $O(n^{10})$

Pb : La coordination à l'aide de cailloux est-elle suffisante ?

Résultats précédents (asynchrone)

Conditions minimales sur la coordination des agents.

Marqueurs d'arêtes, anneaux, liens FIFO

[Dobrev, Santoro, Shi. CIAC 2006]

- $O(1)$ marqueurs d'arêtes en tout, jusqu'à 3 par nœud
- Nb de déplacements : $\Theta(n \log n)$

Marqueurs d'arêtes, graphes arbitraires, liens FIFO

[Dobrev, Flocchini, Kralovic, Santoro. IFIP TCS 2006]

- Aucune information (à part n)
- Nb d'agents : $\Delta + 1$
- Nb de déplacements : env. $O(n^{10})$

Pb : La coordination à l'aide de cailloux est-elle suffisante ?

Résultats précédents (asynchrone)

Conditions minimales sur la coordination des agents.

Marqueurs d'arêtes, anneaux, liens FIFO

[Dobrev, Santoro, Shi. CIAC 2006]

- $O(1)$ marqueurs d'arêtes en tout, jusqu'à 3 par nœud
- Nb de déplacements : $\Theta(n \log n)$

Marqueurs d'arêtes, graphes arbitraires, liens FIFO

[Dobrev, Flocchini, Kralovic, Santoro. IFIP TCS 2006]

- Aucune information (à part n)
- Nb d'agents : $\Delta + 1$
- Nb de déplacements : env. $O(n^{10})$

Pb : La coordination à l'aide de cailloux est-elle suffisante ?

Nos résultats

Modèle

- Information complète (graphes arbitraires)
- Cailloux indistinguables (au plus un par sommet)
- Liens non nécessairement FIFO

Résultat optimal

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ déplacements

Nos résultats

Modèle

- Information complète (graphes arbitraires)
- Cailloux indistinguables (au plus un par sommet)
- Liens non nécessairement FIFO

Résultat optimal

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ déplacements

Cas de l'anneau

Modèle

- Deux agents identiques équipés chacun d'un caillou
- Cailloux identiques
- Au plus un caillou sur un sommet / transporté par un agent
- Anneau asynchrone contenant exactement un trou noir

Définition

- Arête de plus petit port de la base de départ : droite
- Autre direction : gauche

Cas de l'anneau

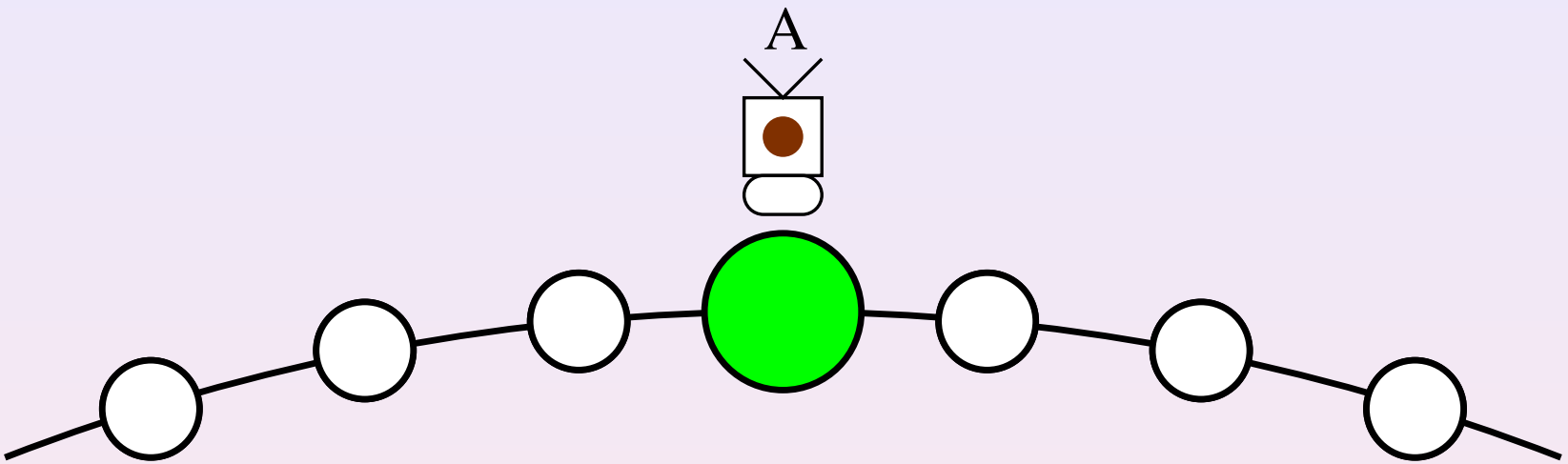
Modèle

- Deux agents identiques équipés chacun d'un caillou
- Cailloux identiques
- Au plus un caillou sur un sommet / transporté par un agent
- Anneau asynchrone contenant exactement un trou noir

Définition

- Arête de plus petit port de la base de départ : droite
- Autre direction : gauche

Premiers pas

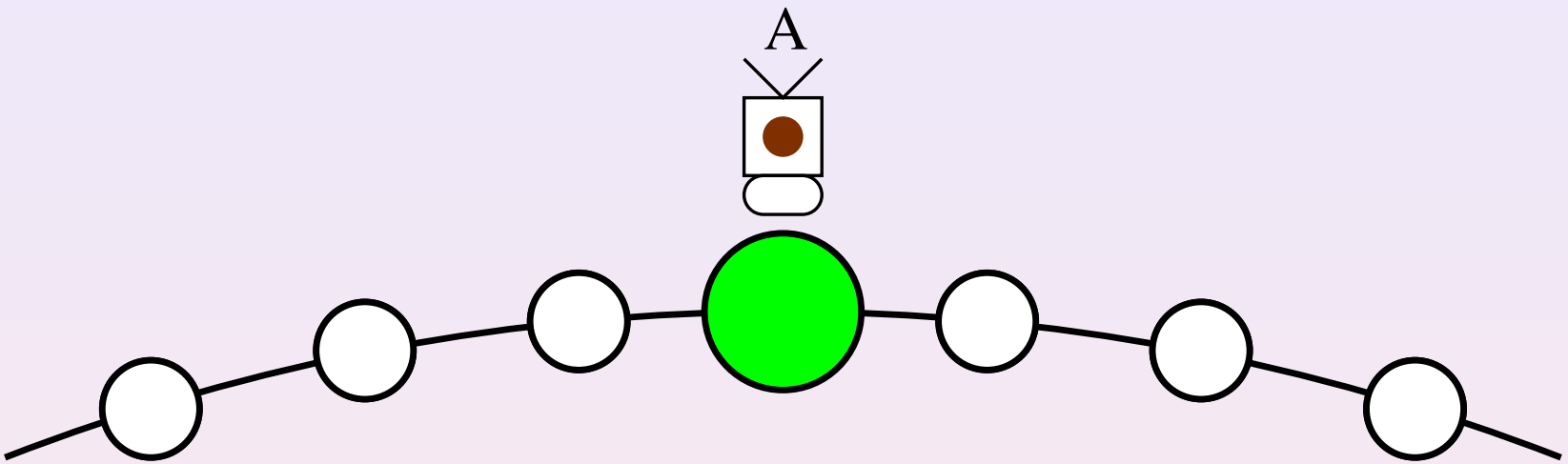


Début de l'algorithme

If nœud vide **then**

Aller à droite

Premiers pas



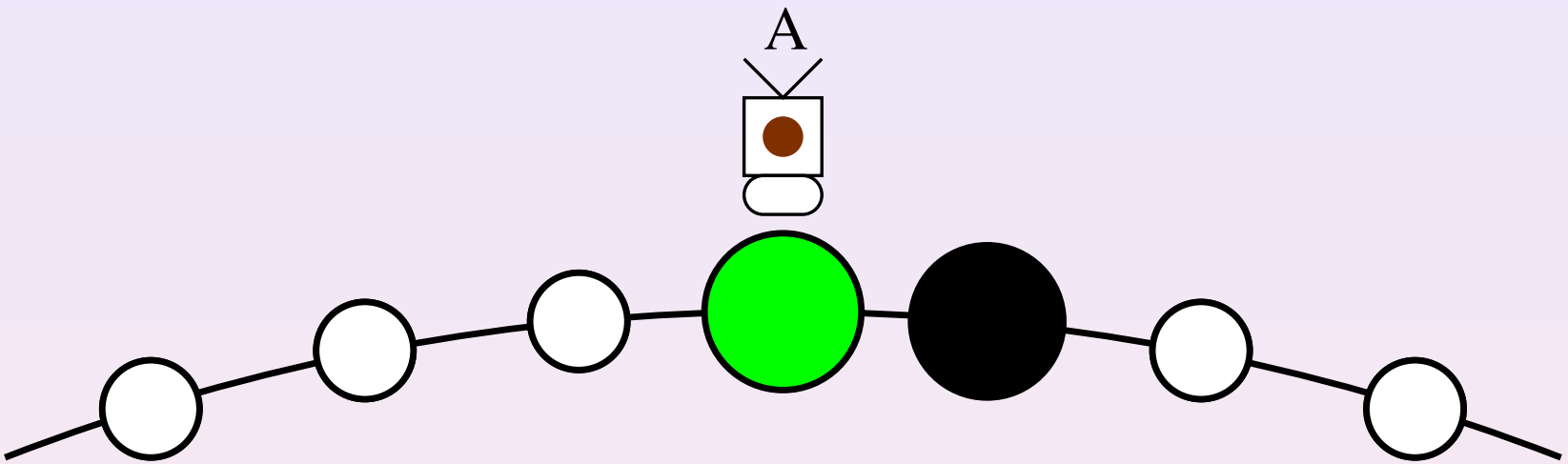
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



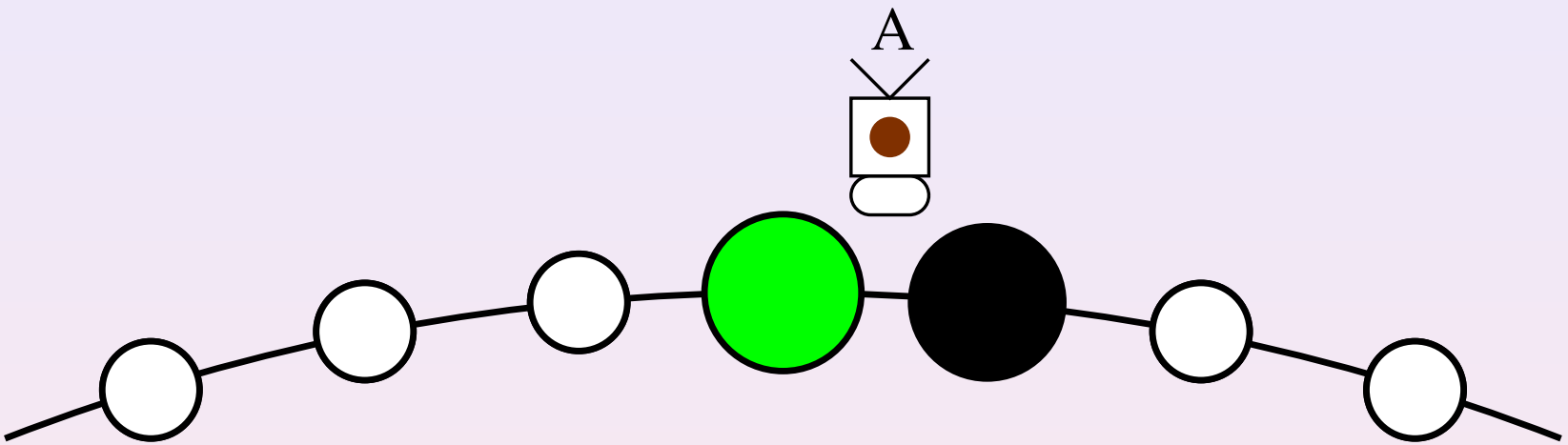
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



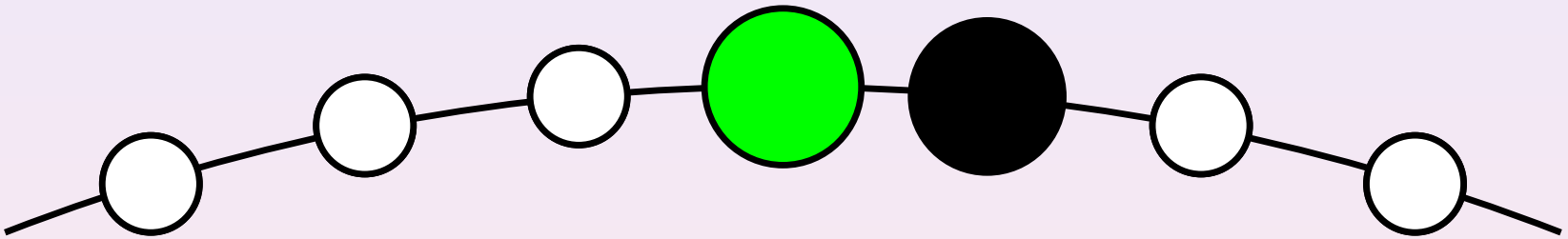
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



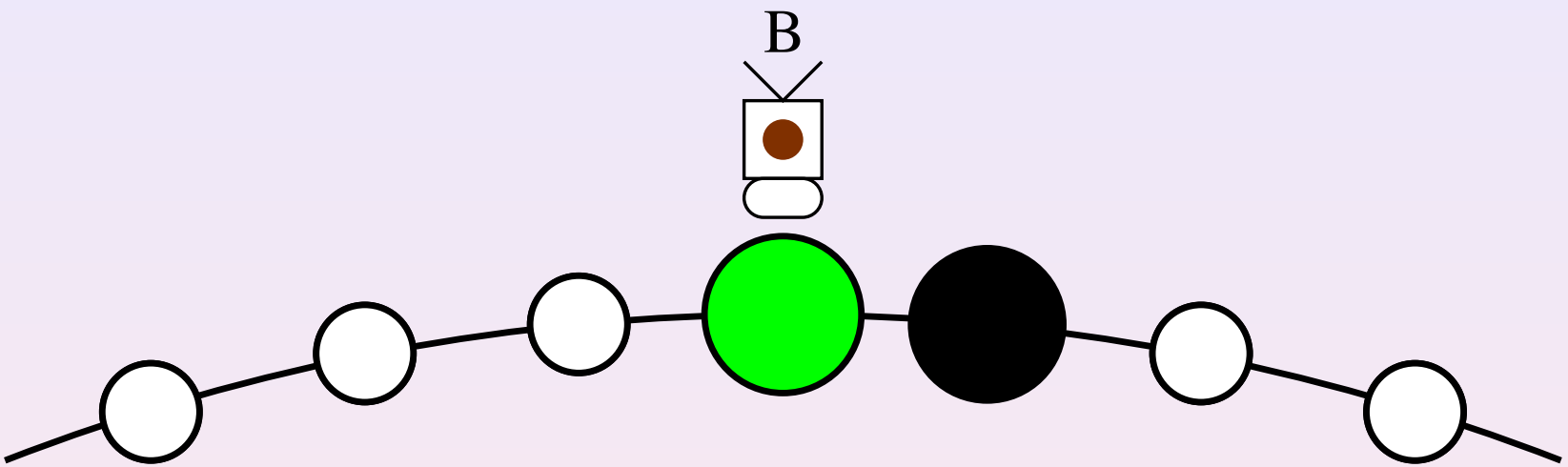
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



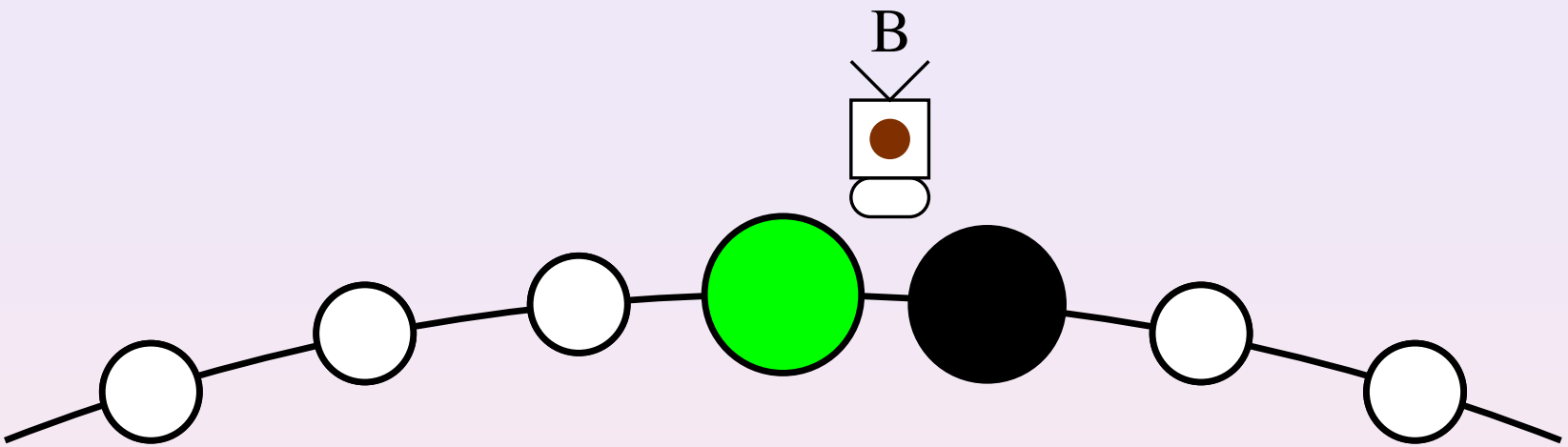
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



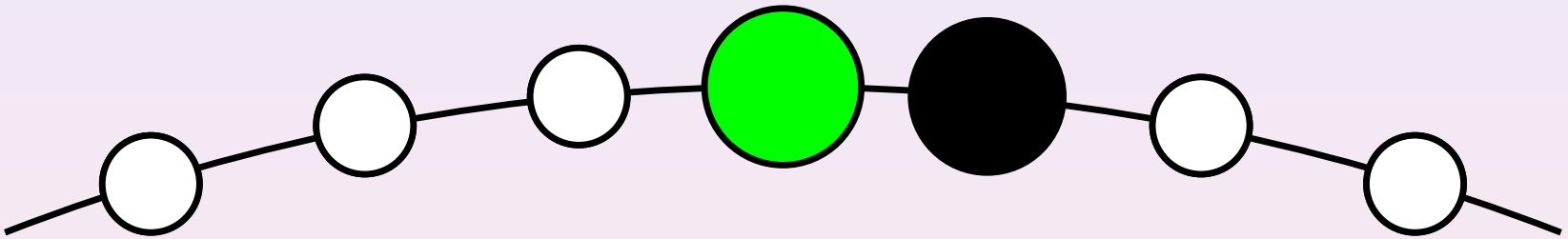
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



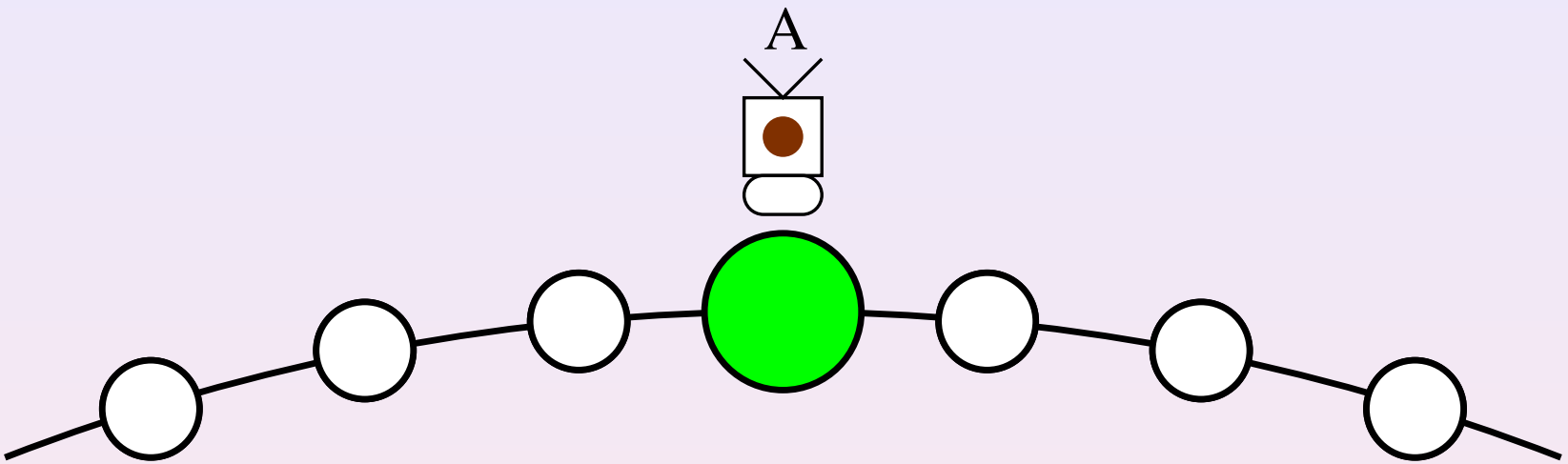
Début de l'algorithme

If nœud vide **then**

Ne pas déposer le caillou

Aller à droite

Premiers pas



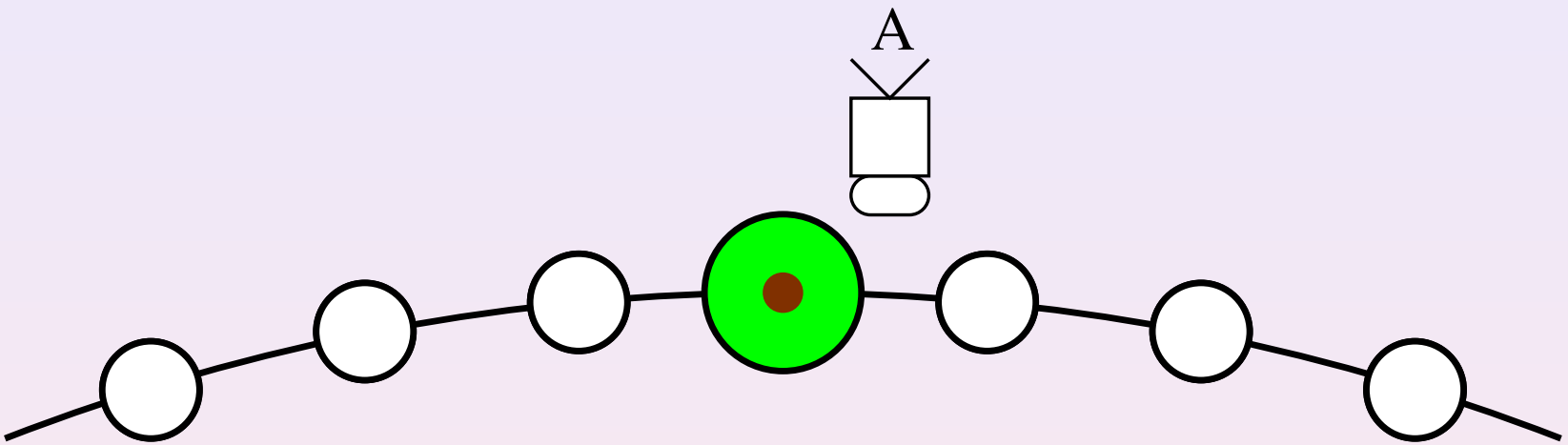
Début de l'algorithme

If nœud vide **then**

Déposer le caillou

Aller à droite

Premiers pas



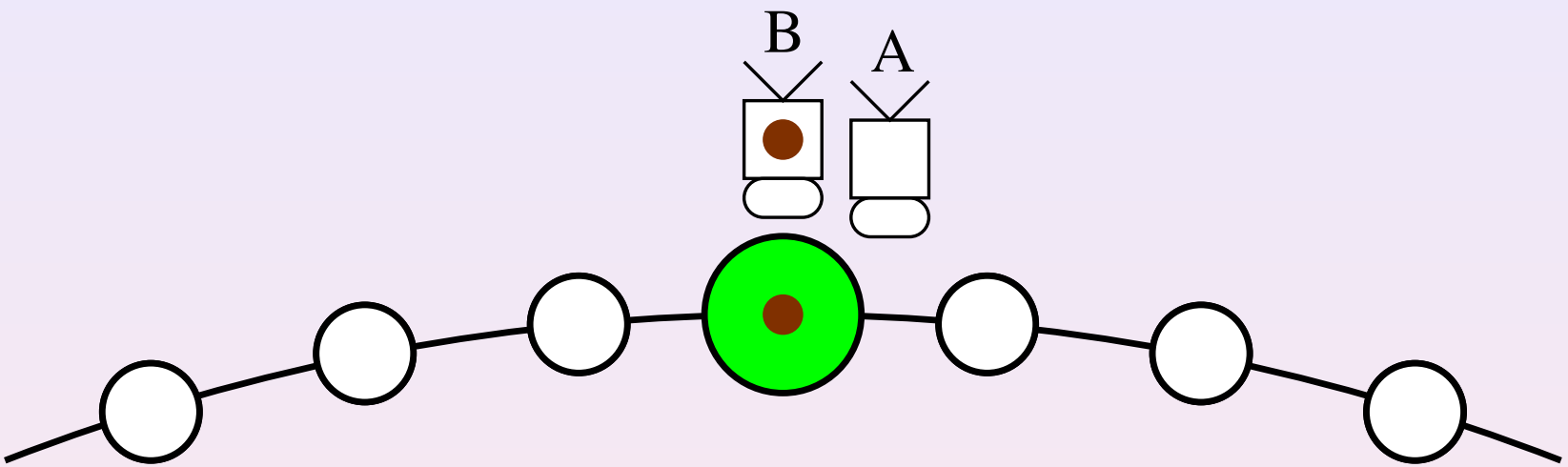
Début de l'algorithme

If nœud vide **then**

Déposer le caillou

Aller à droite

Premiers pas



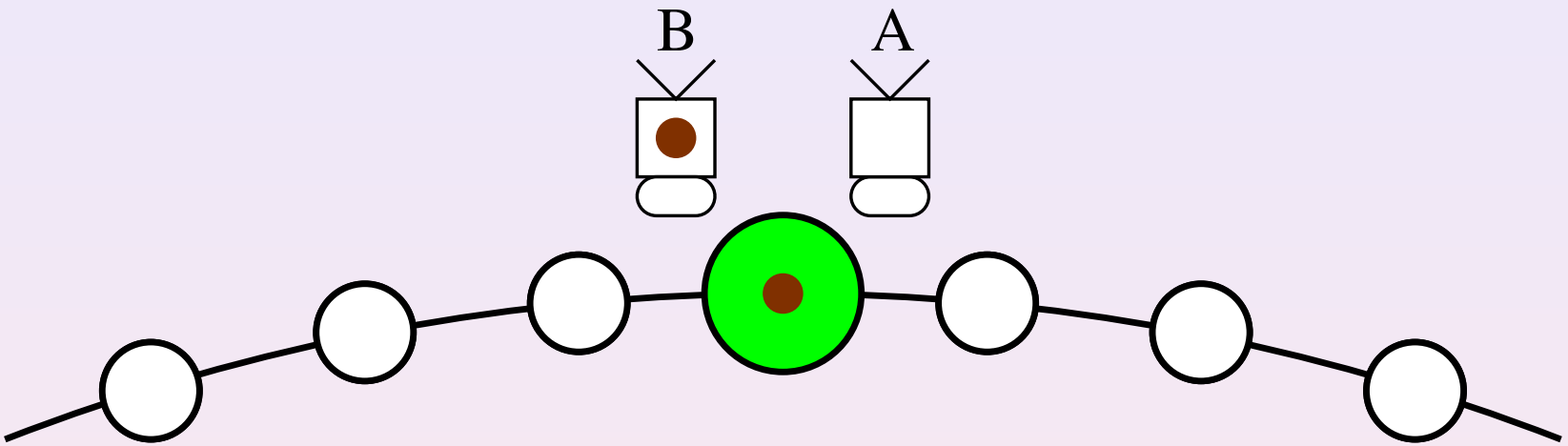
Début de l'algorithme

If nœud vide **then**

Déposer le caillou

Aller à droite

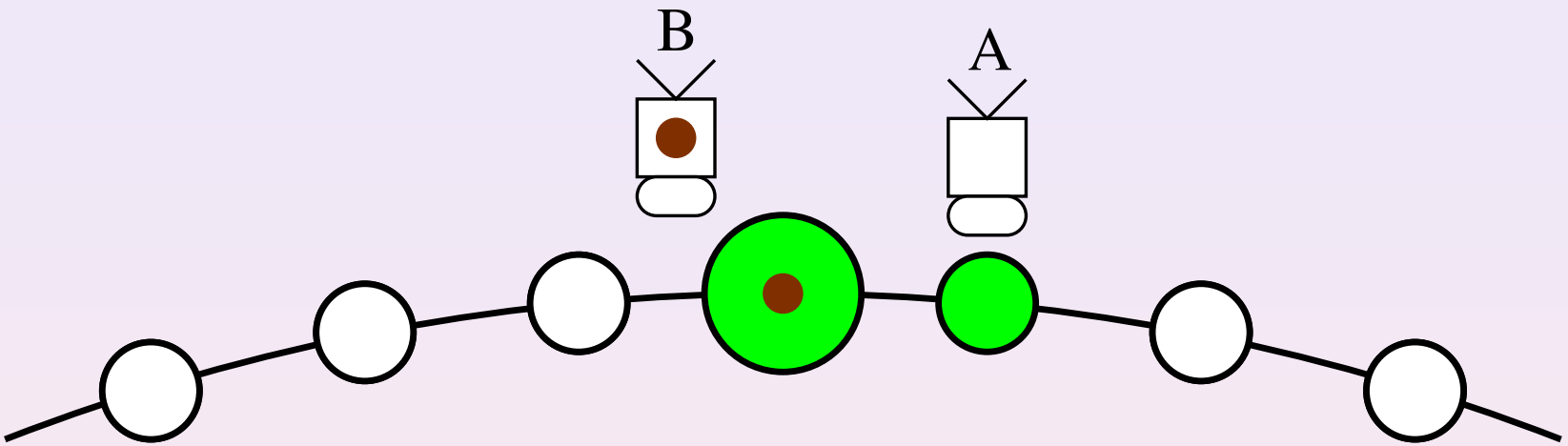
Premiers pas



Début de l'algorithme

If nœud vide **then**
    Déposer le caillou
    Aller à droite
else Aller à gauche **endif**

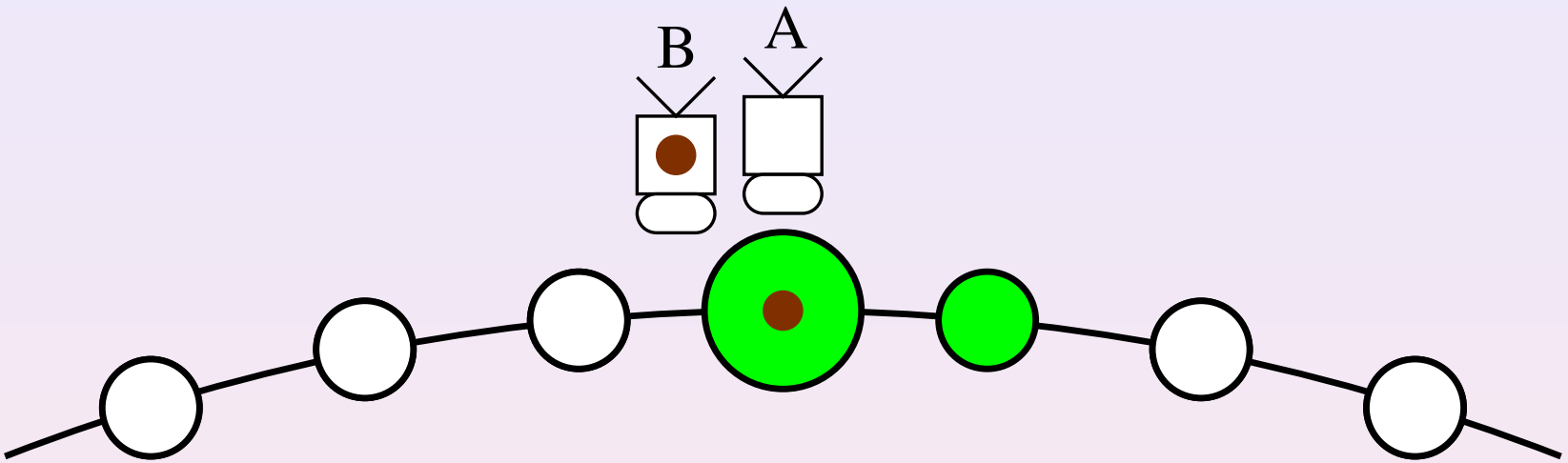
Premiers pas



Début de l'algorithme

If nœud vide **then**
    Déposer le caillou
    Aller à droite
else Aller à gauche **endif**

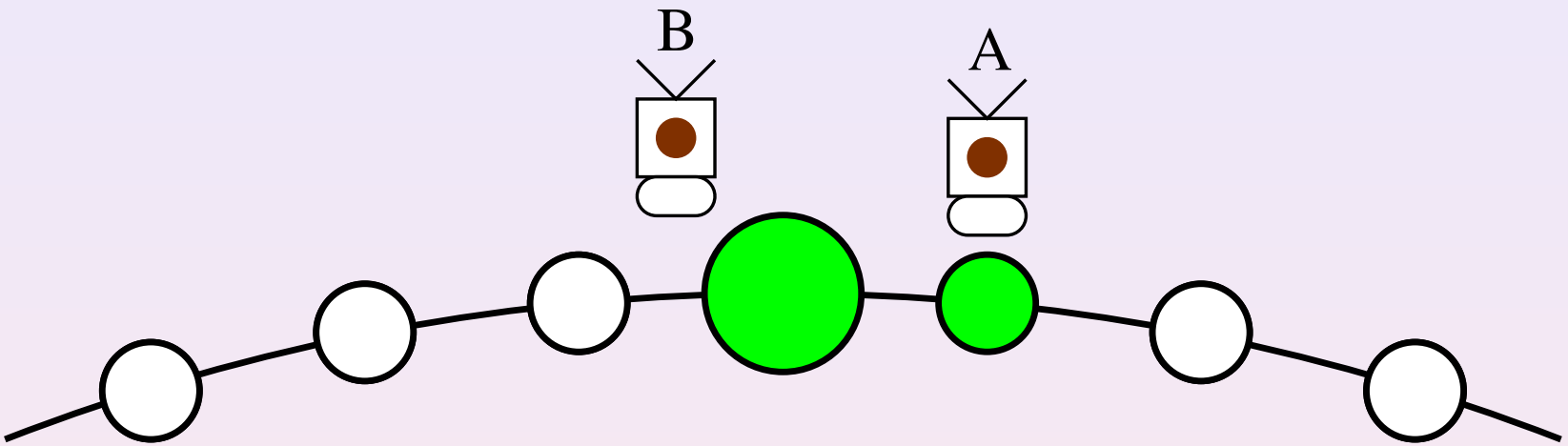
Premiers pas



Début de l'algorithme

If nœud vide **then**
    Déposer le caillou
    Aller à droite
else Aller à gauche **endif**

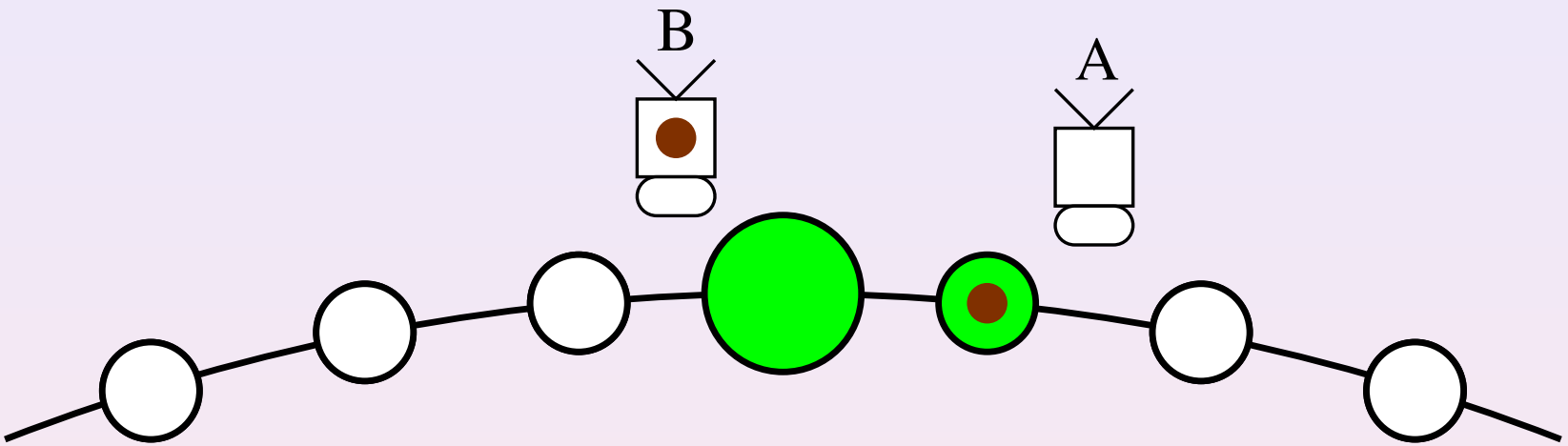
Premiers pas



Début de l'algorithme

If nœud vide **then**
    Déposer le caillou
    Aller à droite
else Aller à gauche **endif**

Premiers pas



Début de l'algorithme

If nœud vide **then**
    Déposer le caillou
    Aller à droite
else Aller à gauche **endif**

Progression prudente

Exploration de $e = \{u, v\}$ de u vers v

Progression prudente simple

Initialement : agent en u avec caillou, nœud sans caillou

- Déposer le caillou (= prévenir du danger) et aller en v
- Revenir au sommet u
- Récupérer le caillou et retourner en v

Progression prudente double

Initialement : agent en u avec caillou, nœud avec caillou

- Aller en v
- Déposer le caillou et revenir en u
- Récupérer l'autre caillou et retourner en v

Progression prudente

Exploration de $e = \{u, v\}$ de u vers v

Progression prudente simple

Initialement : agent en u avec caillou, nœud sans caillou

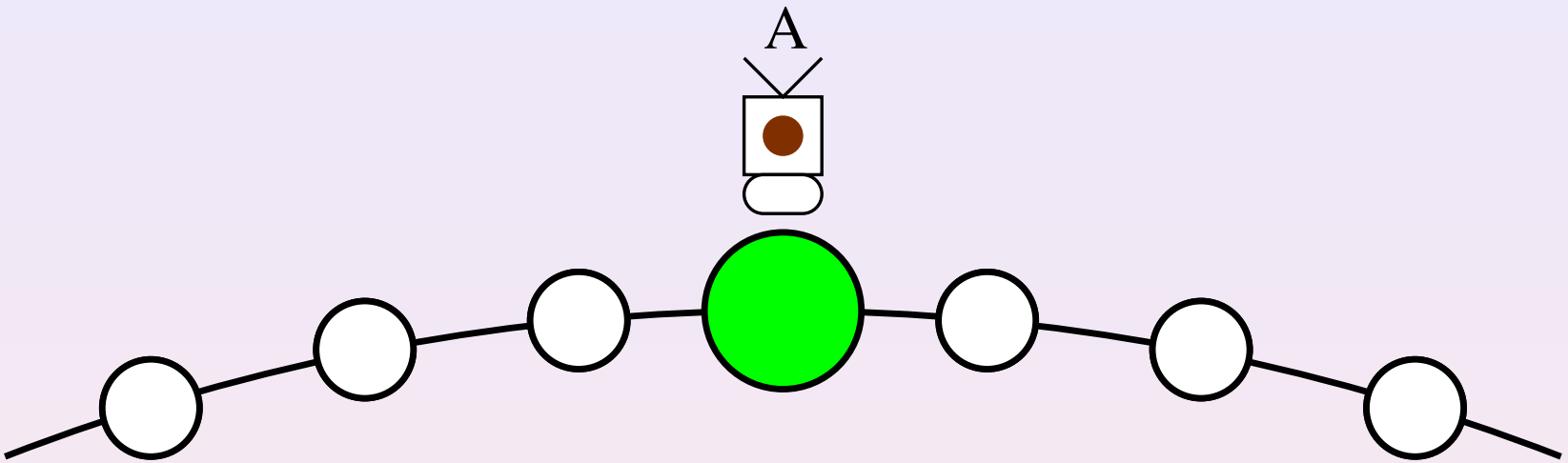
- Déposer le caillou (= prévenir du danger) et aller en v
- Revenir au sommet u
- Récupérer le caillou et retourner en v

Progression prudente double

Initialement : agent en u avec caillou, nœud avec caillou

- Aller en v
- Déposer le caillou et revenir en u
- Récupérer l'autre caillou et retourner en v

Algorithme Ping Pong

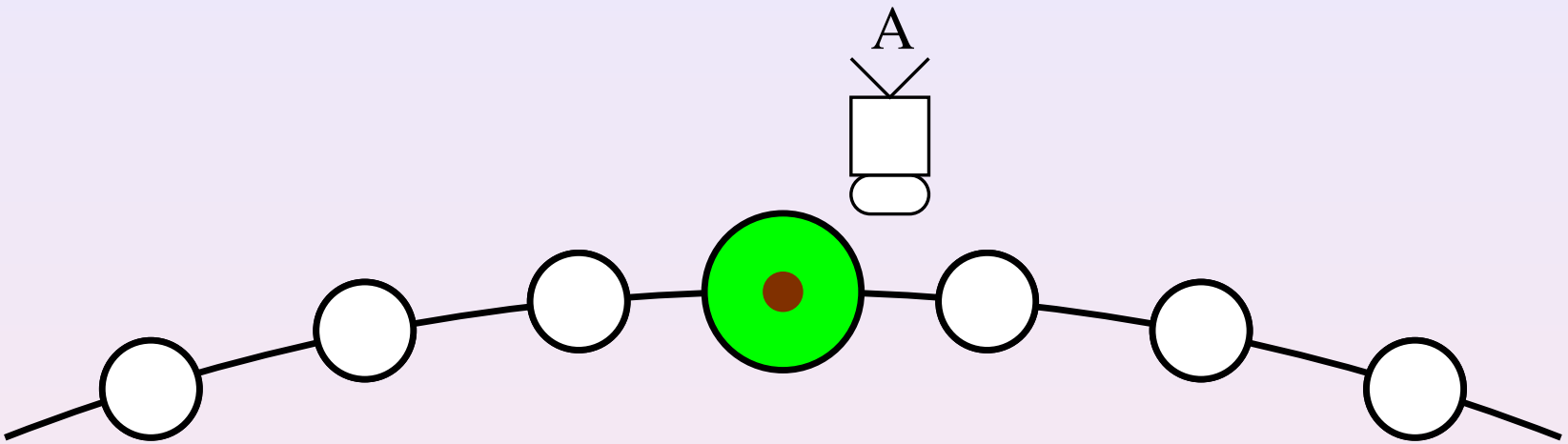


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

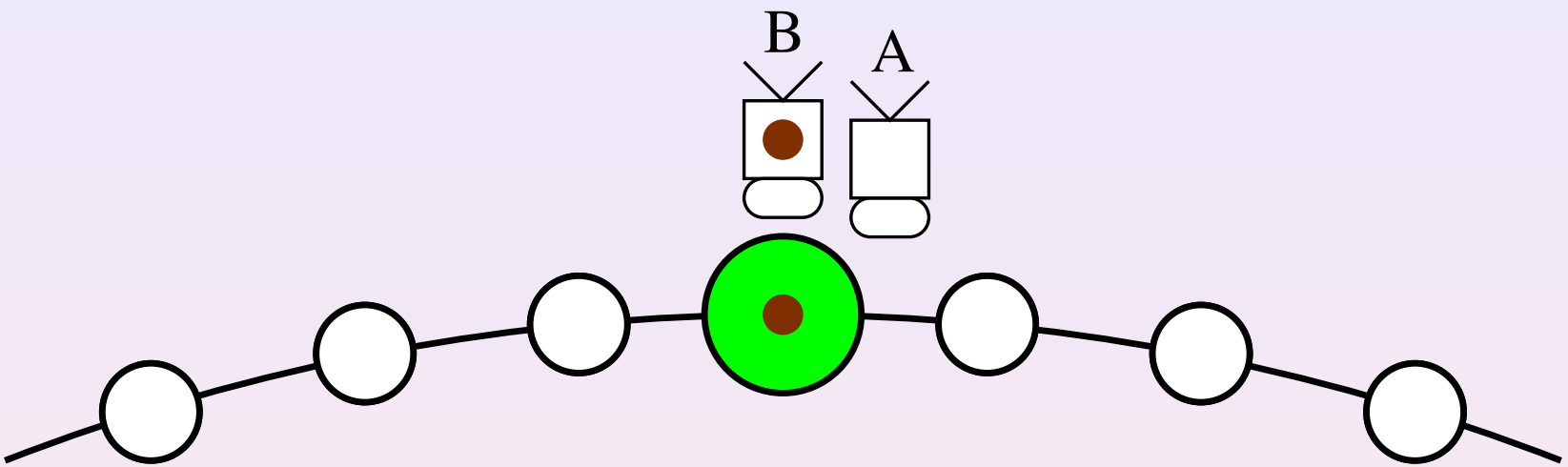


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

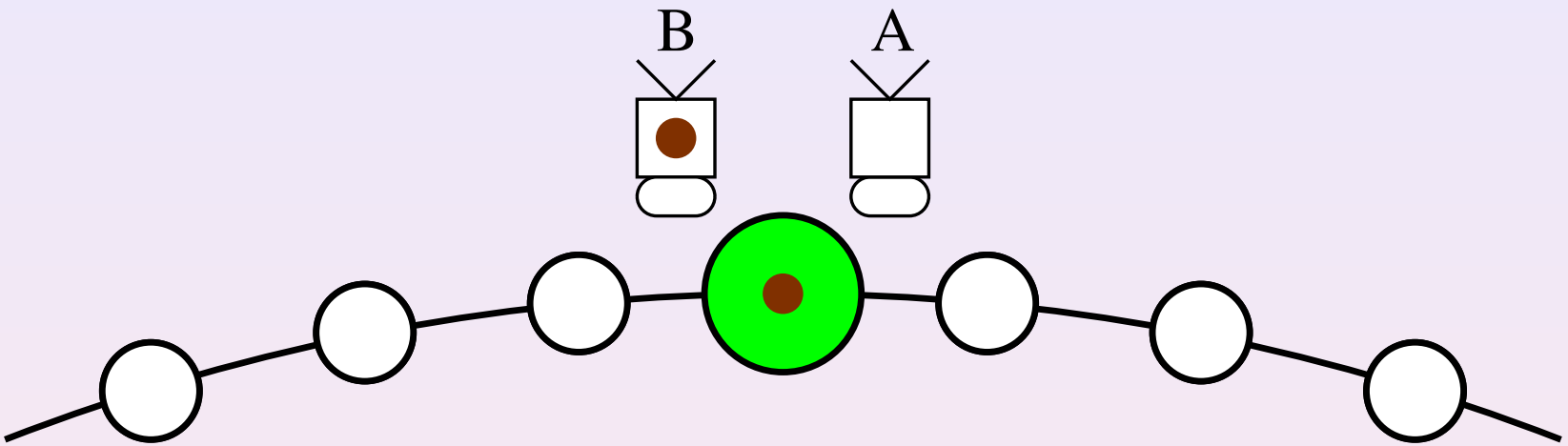


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

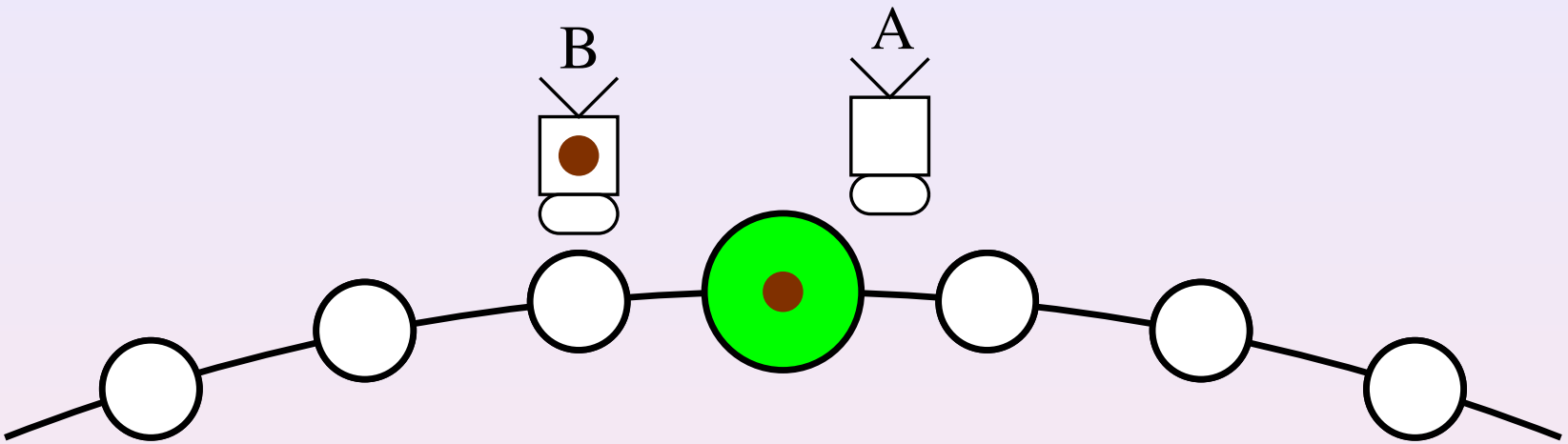


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

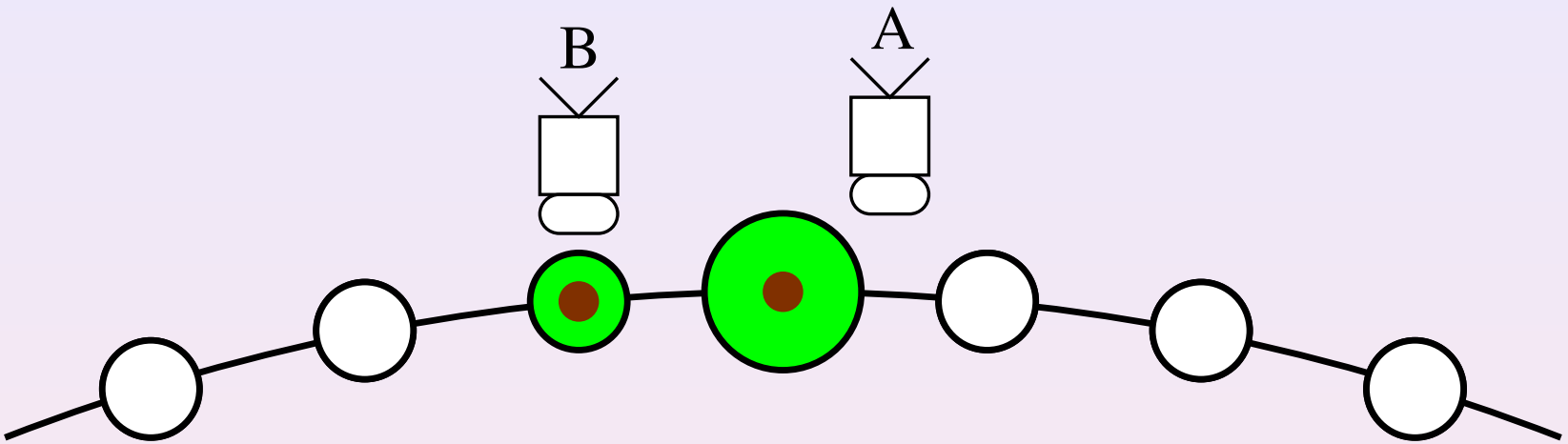


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

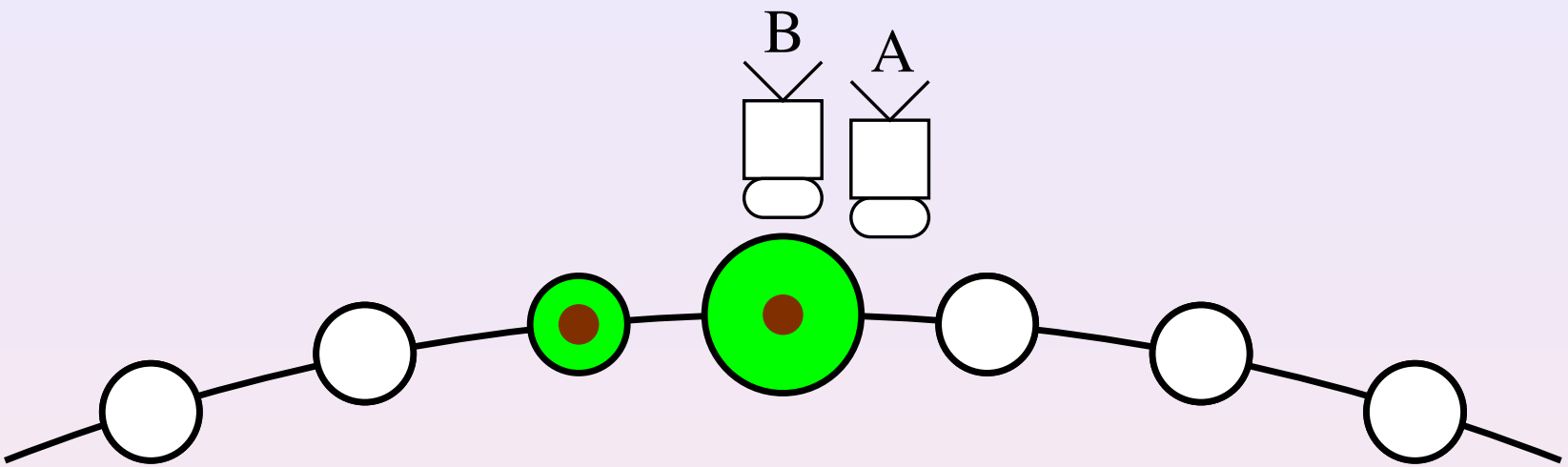


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

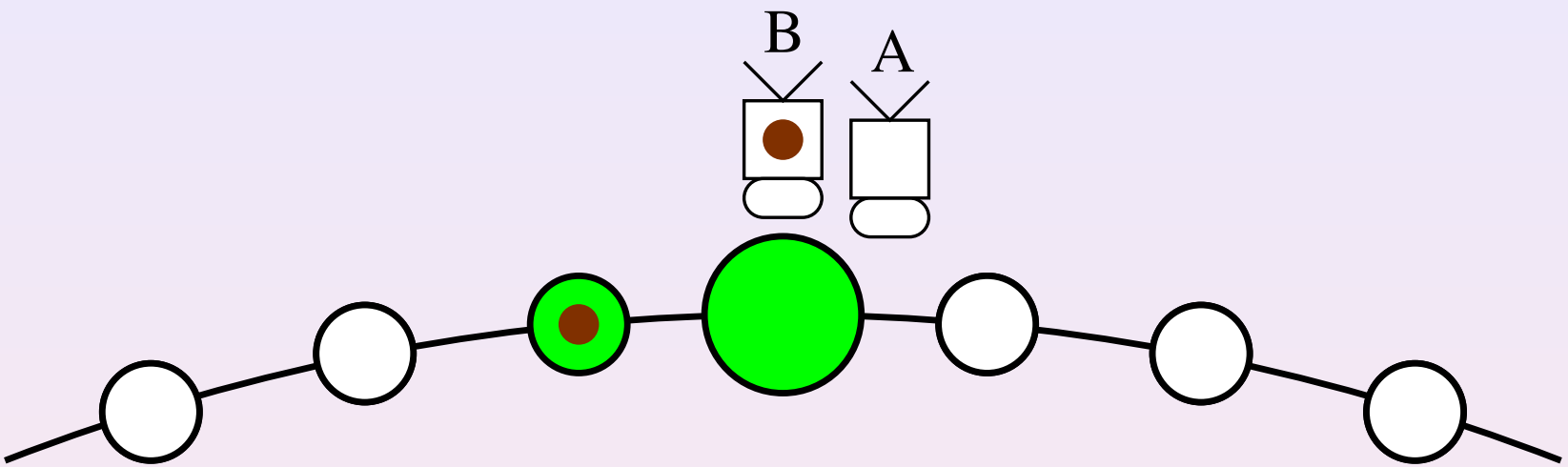


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

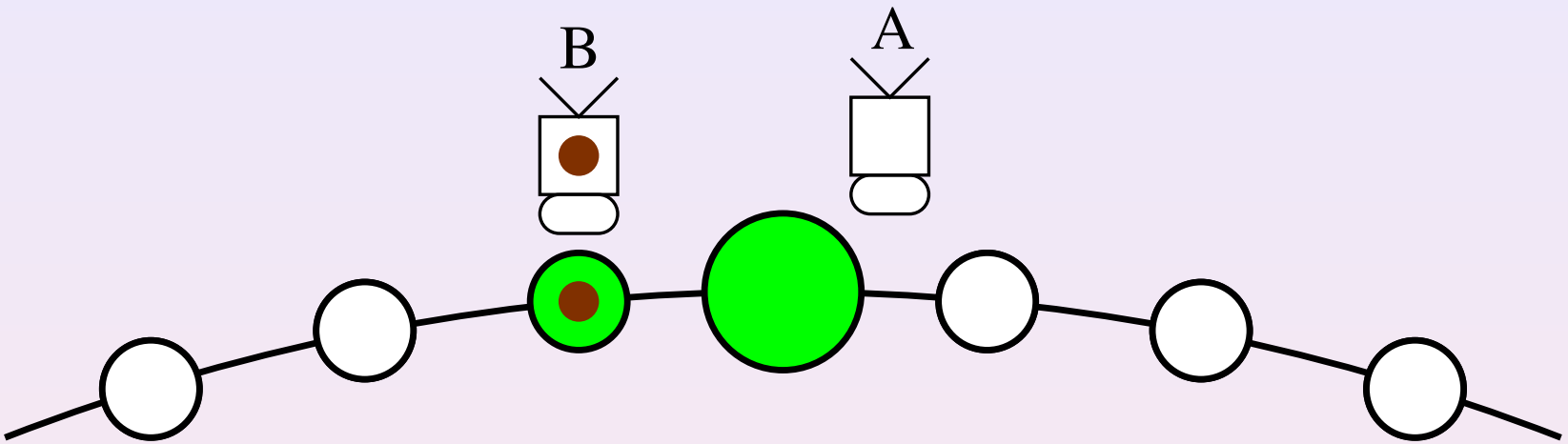


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

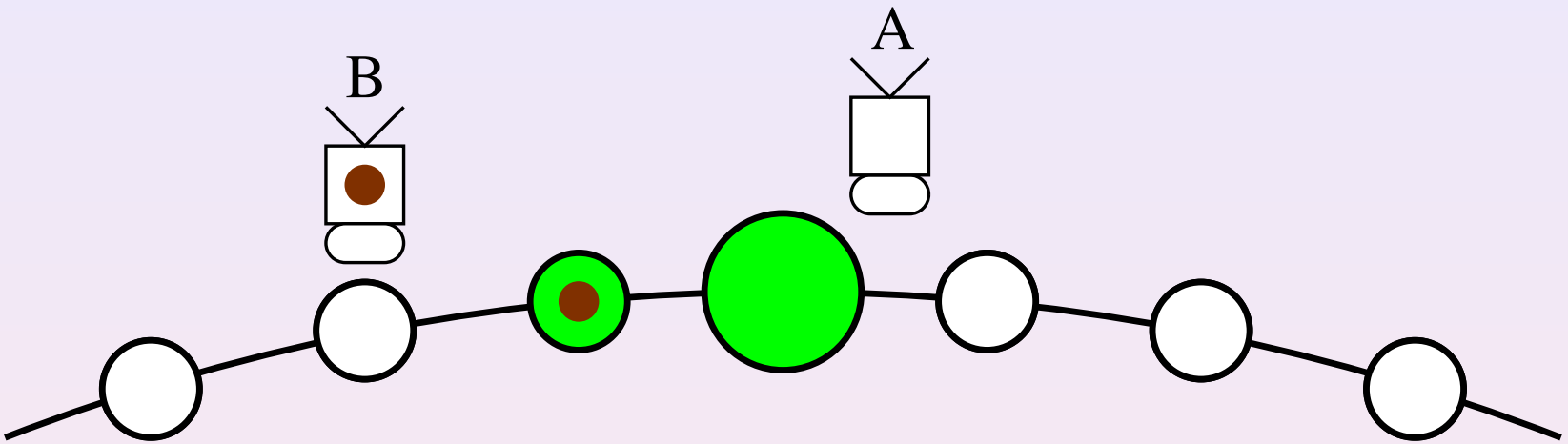


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

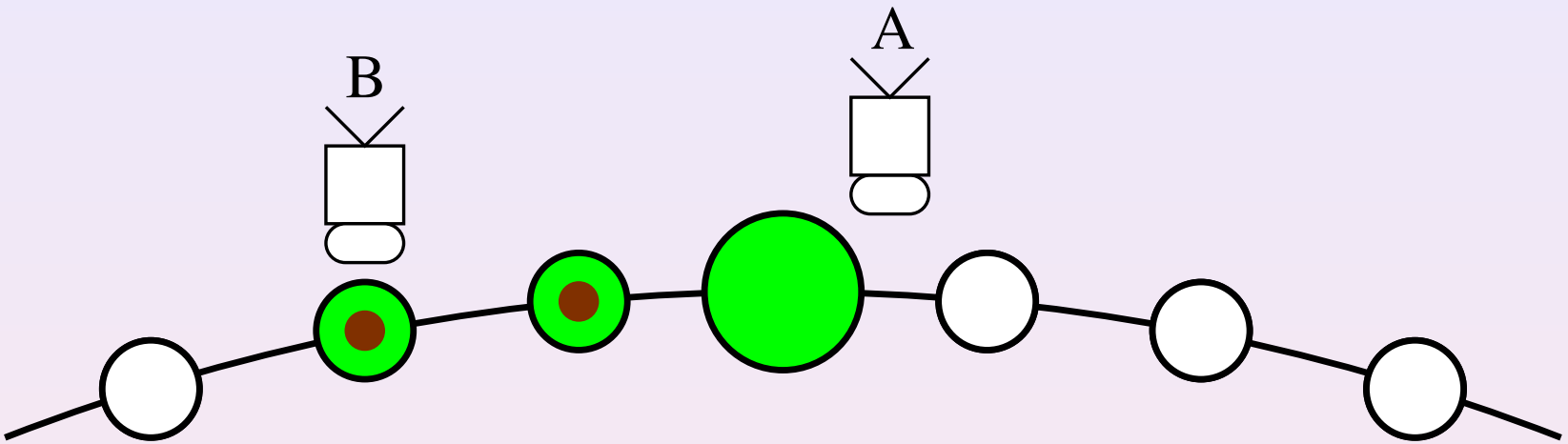


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

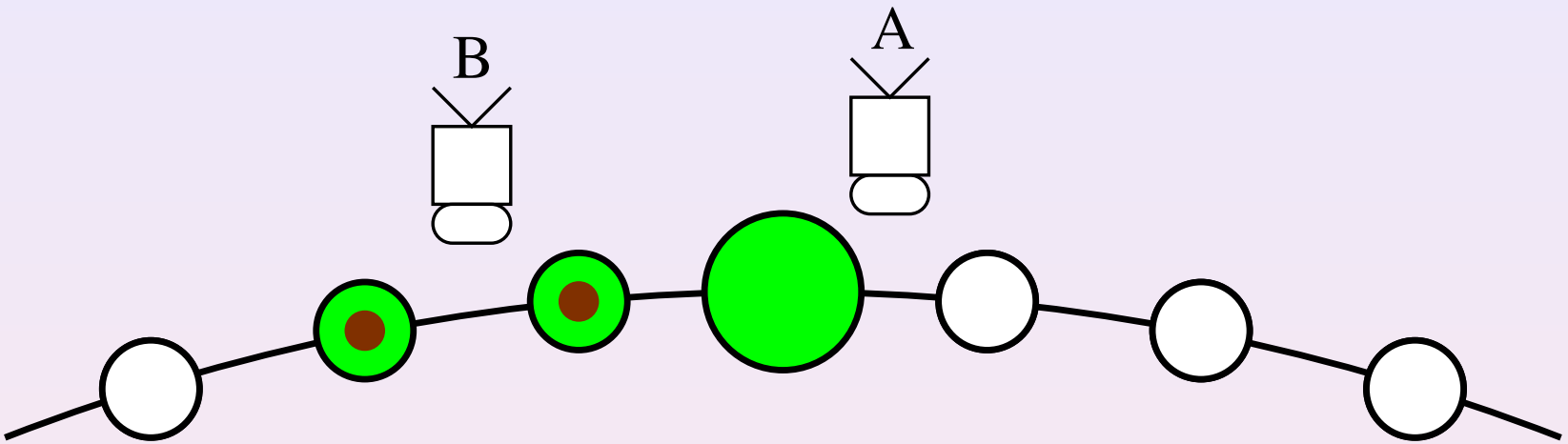


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

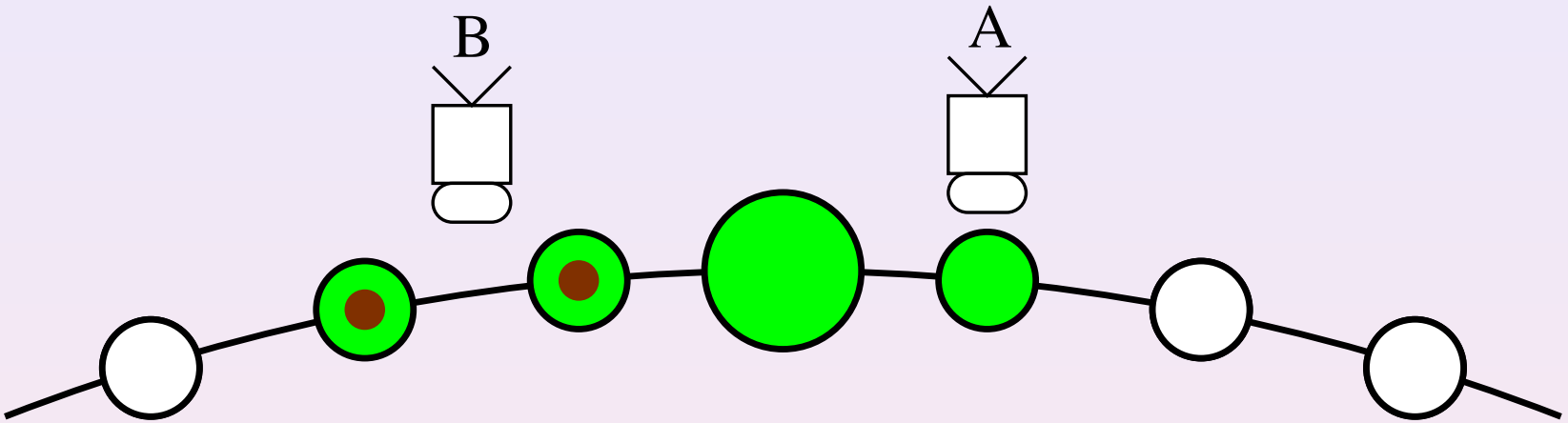


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

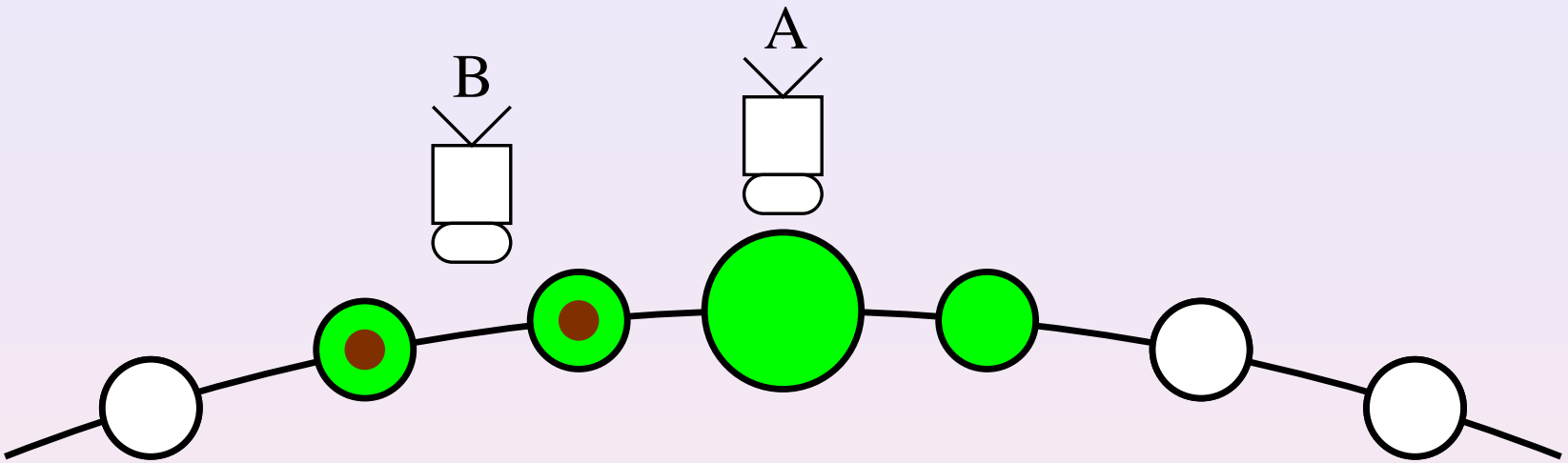


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

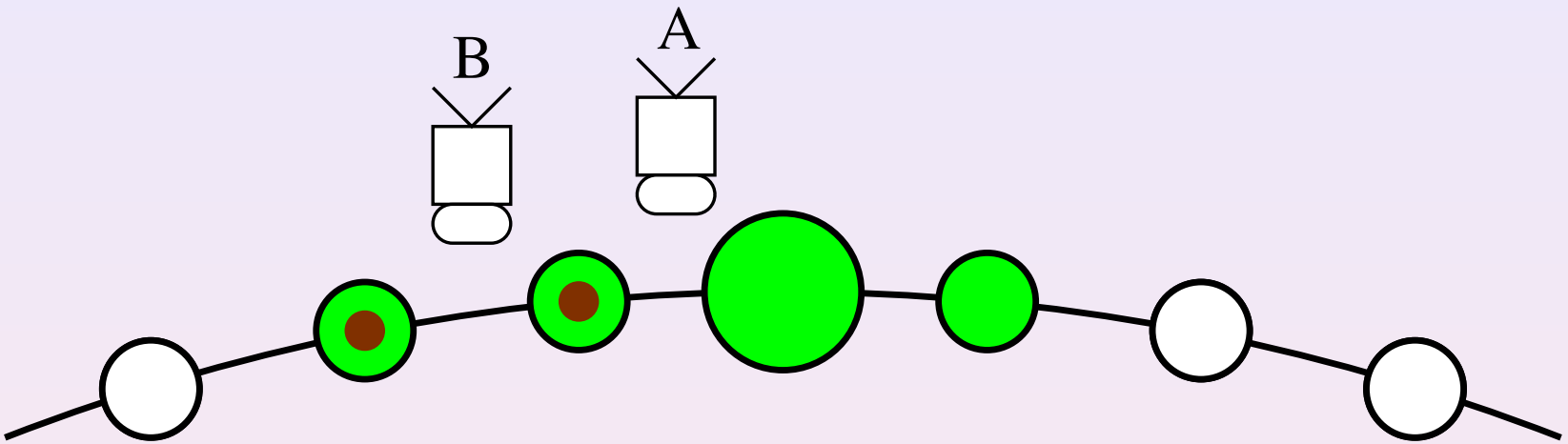


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

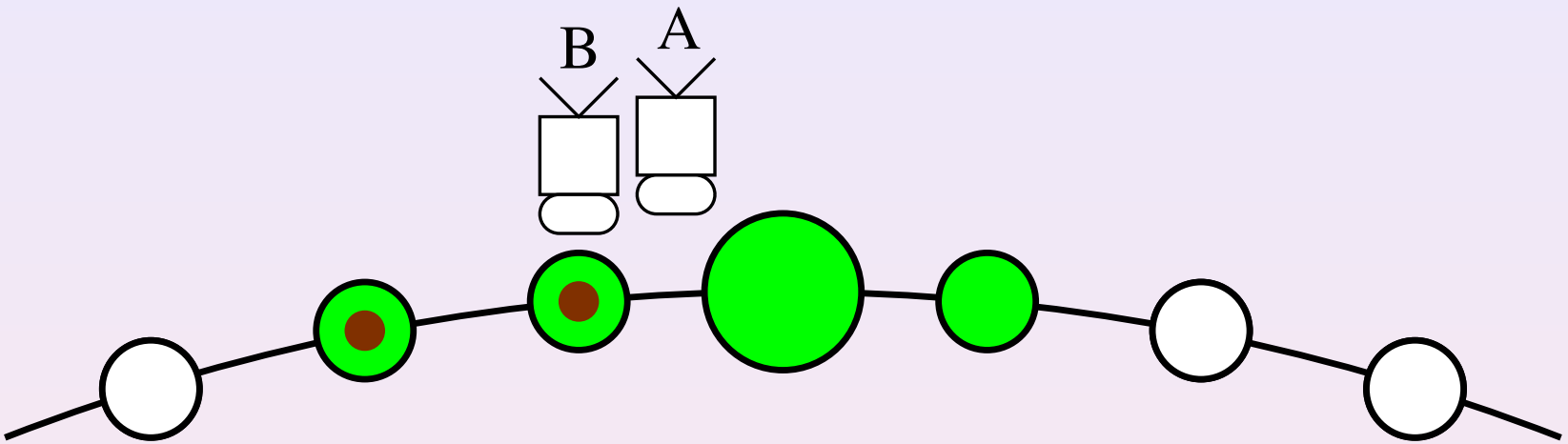


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

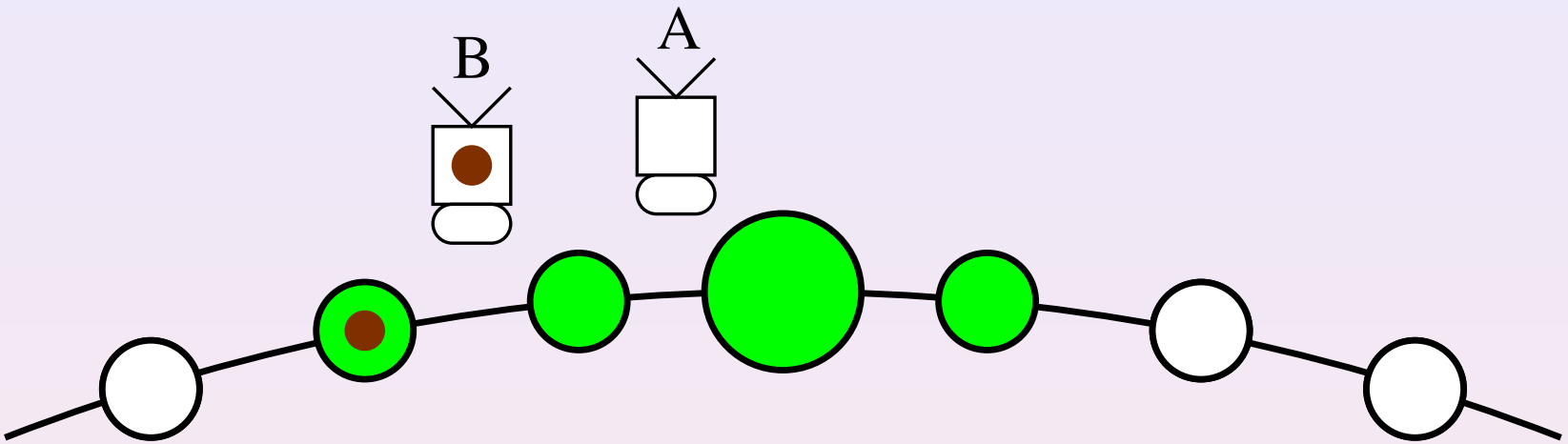


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

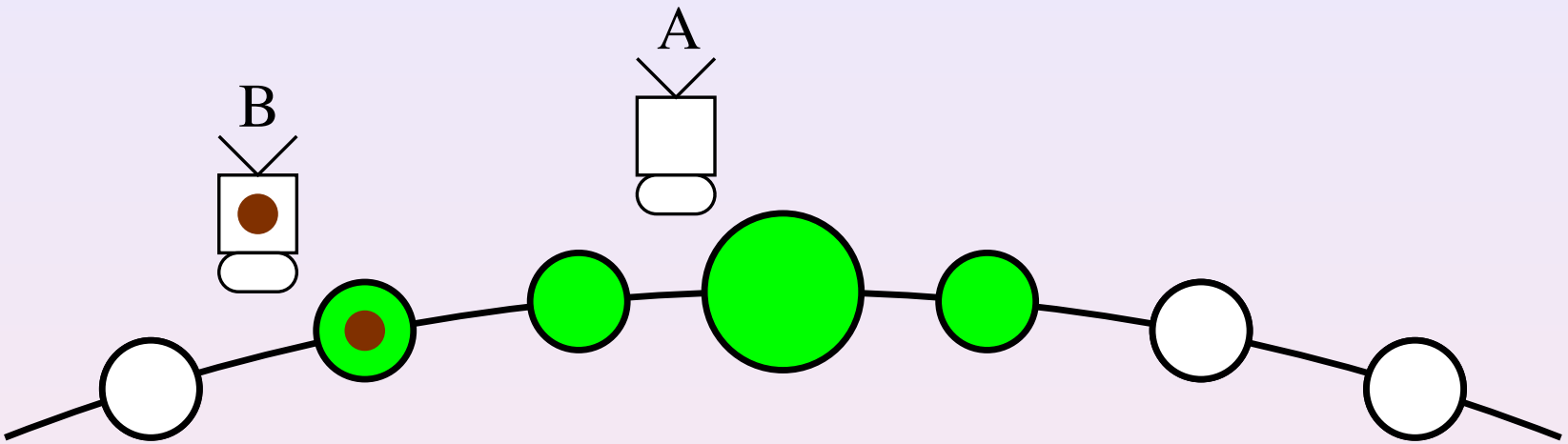


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

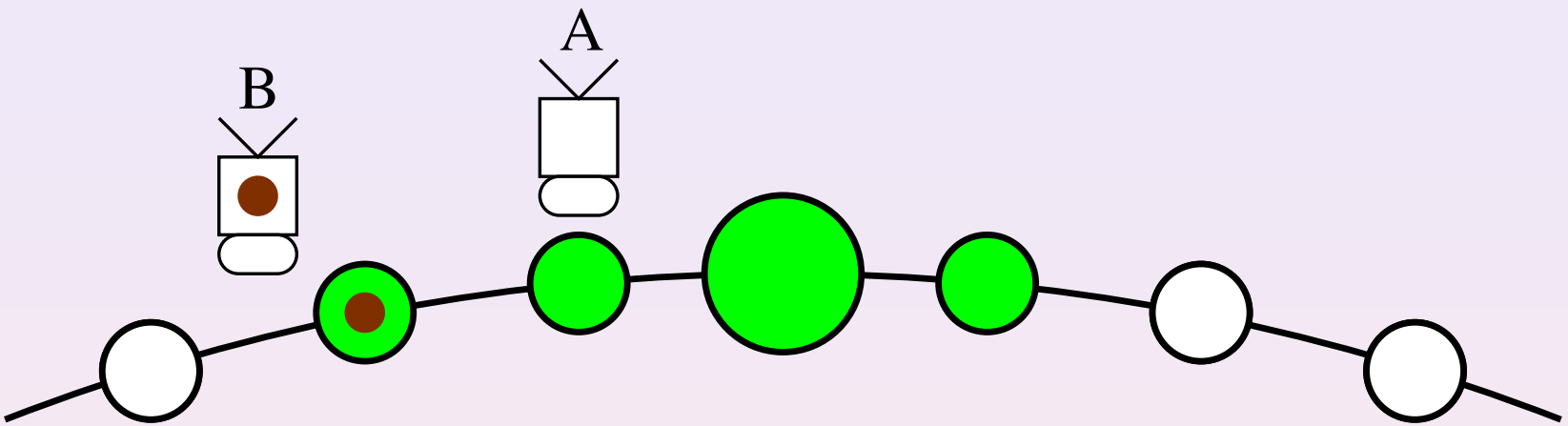


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

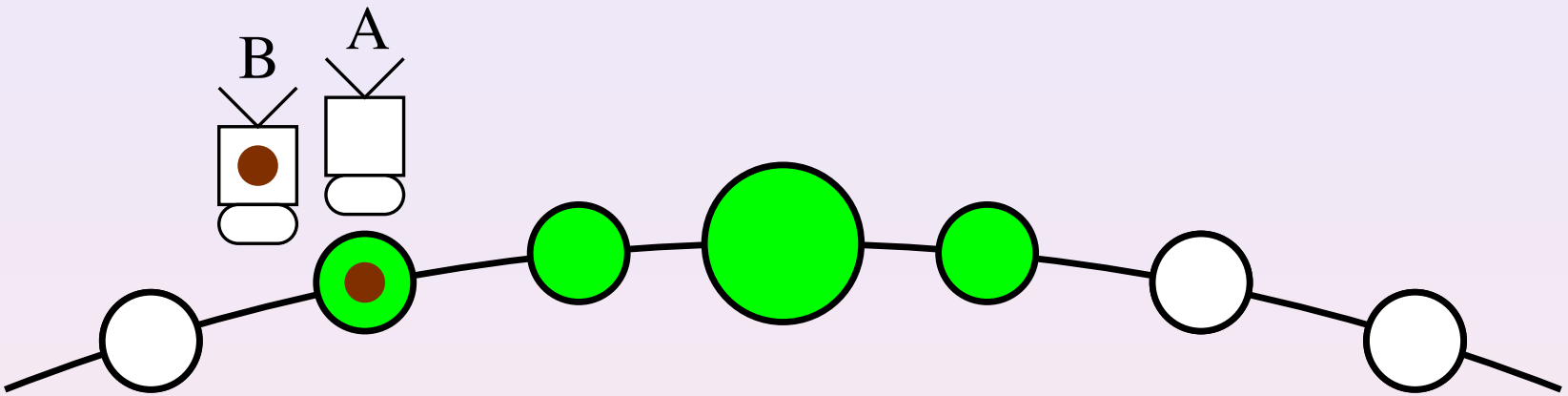


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

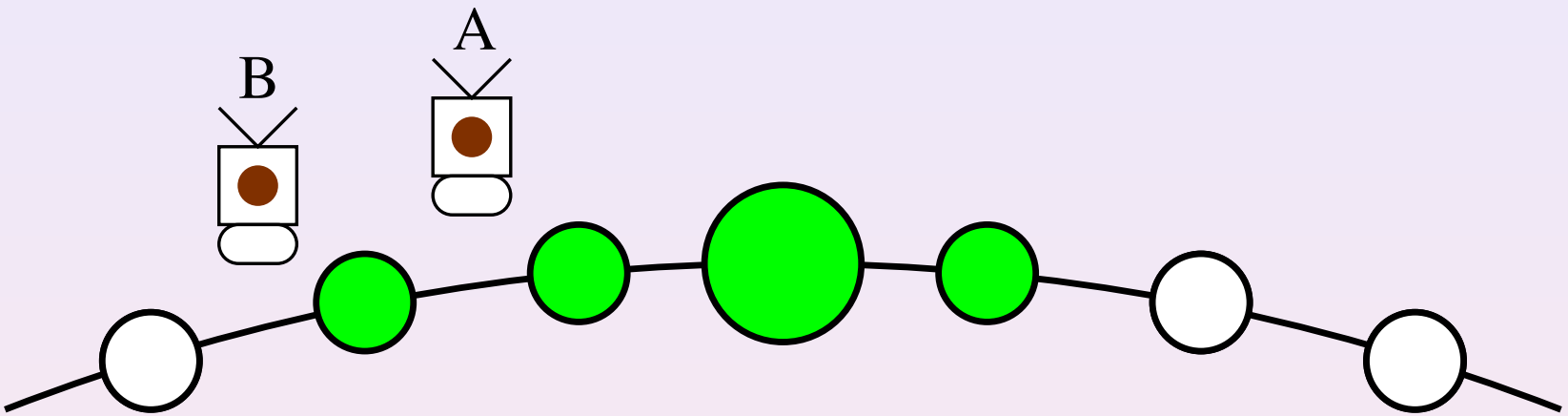


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

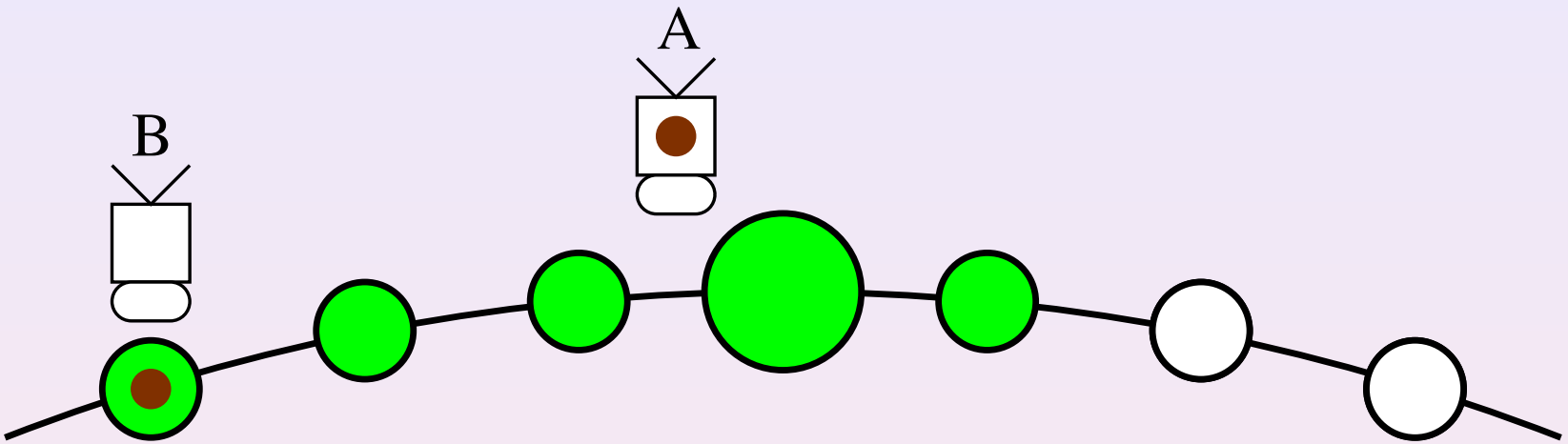


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

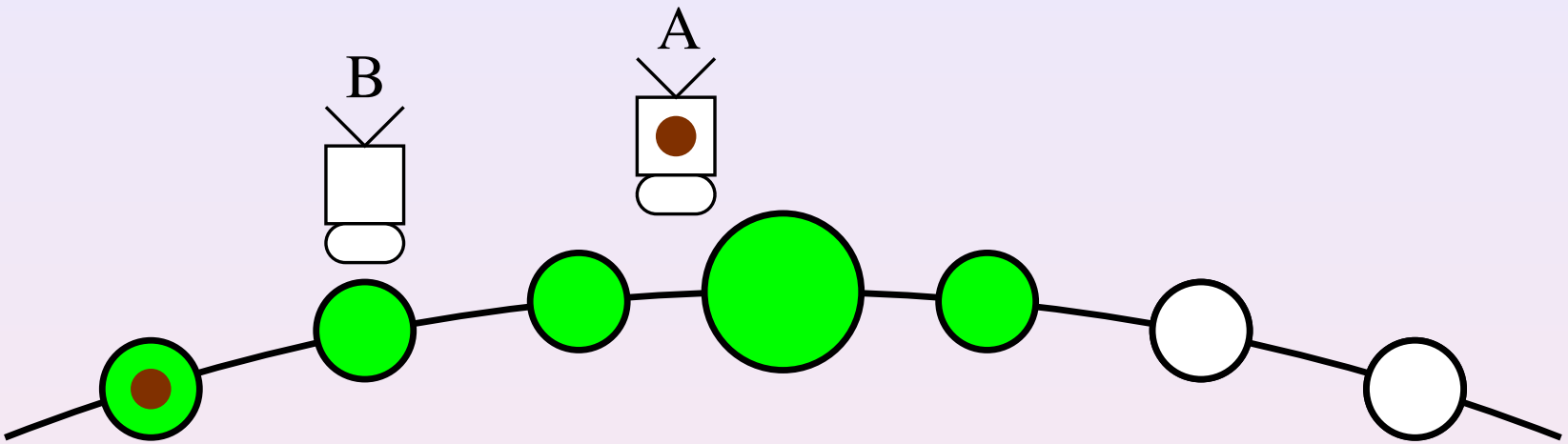


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

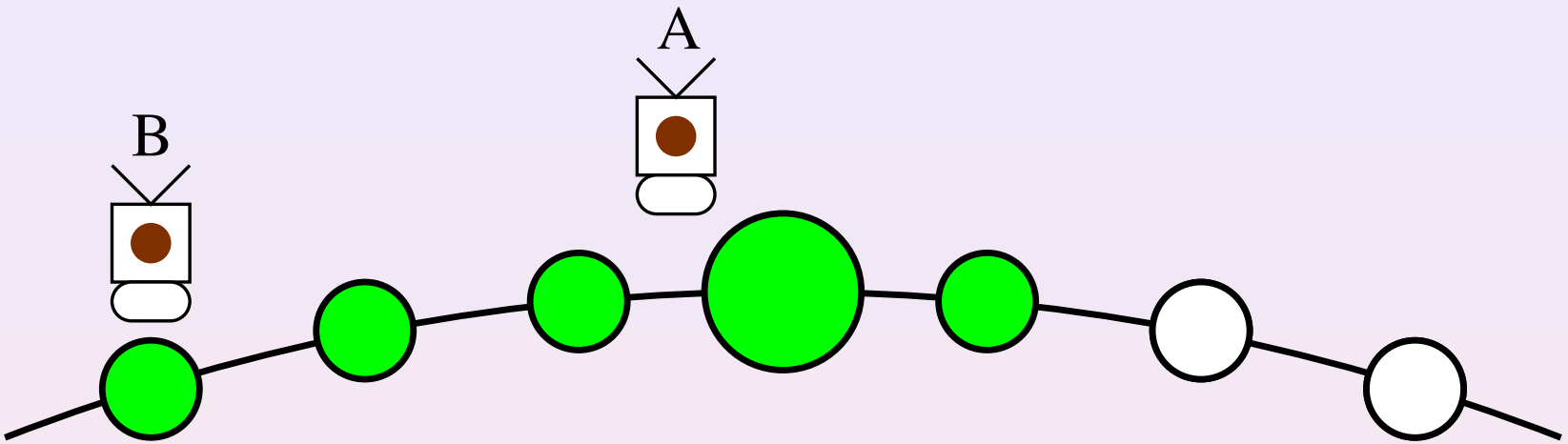


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

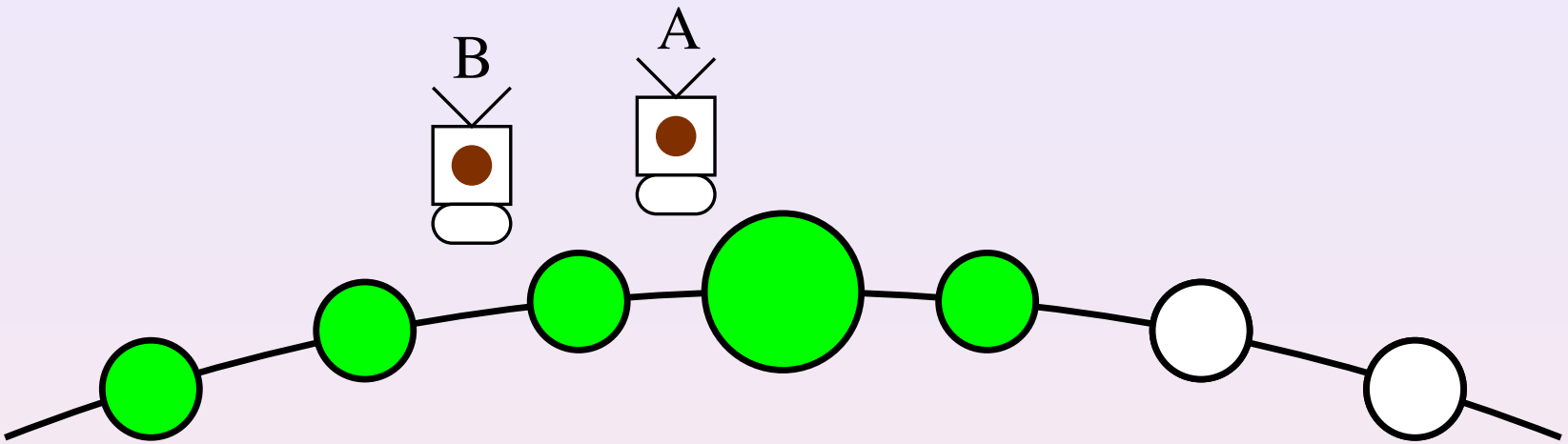


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

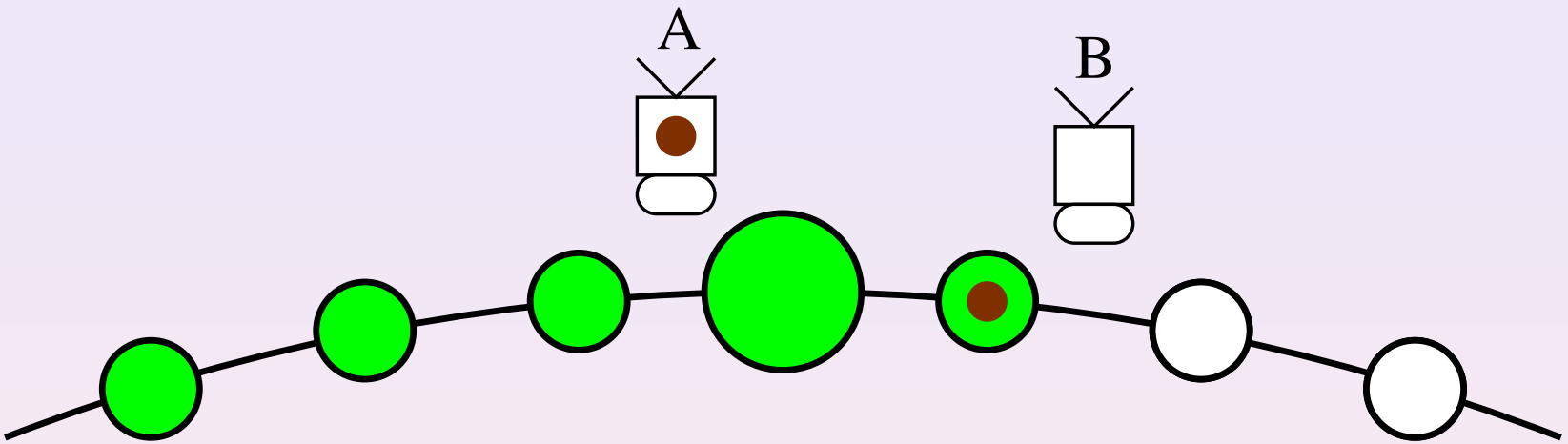


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

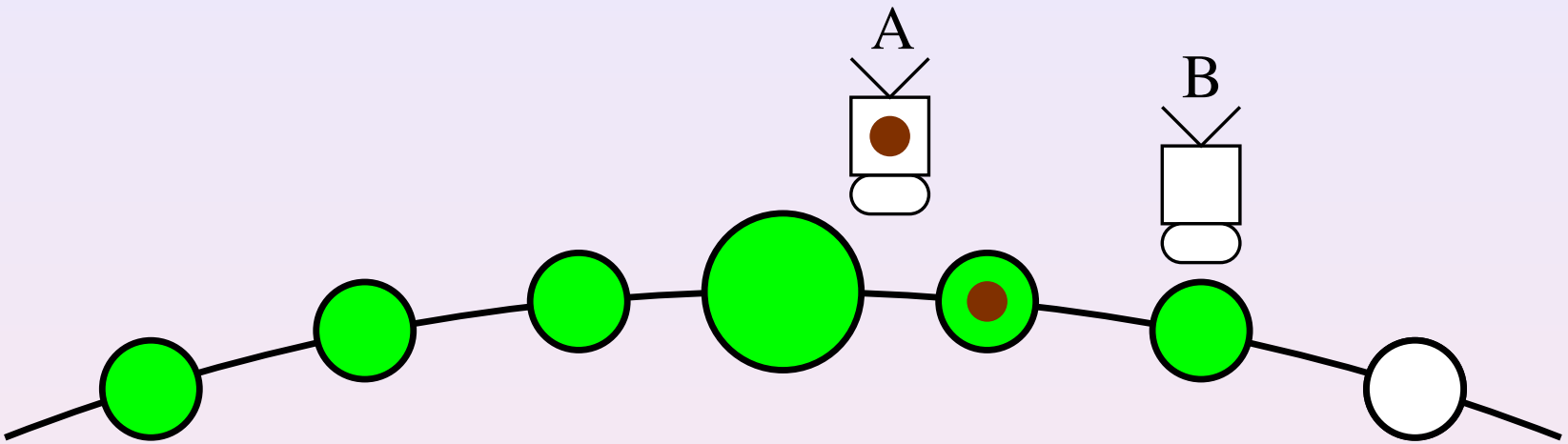


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

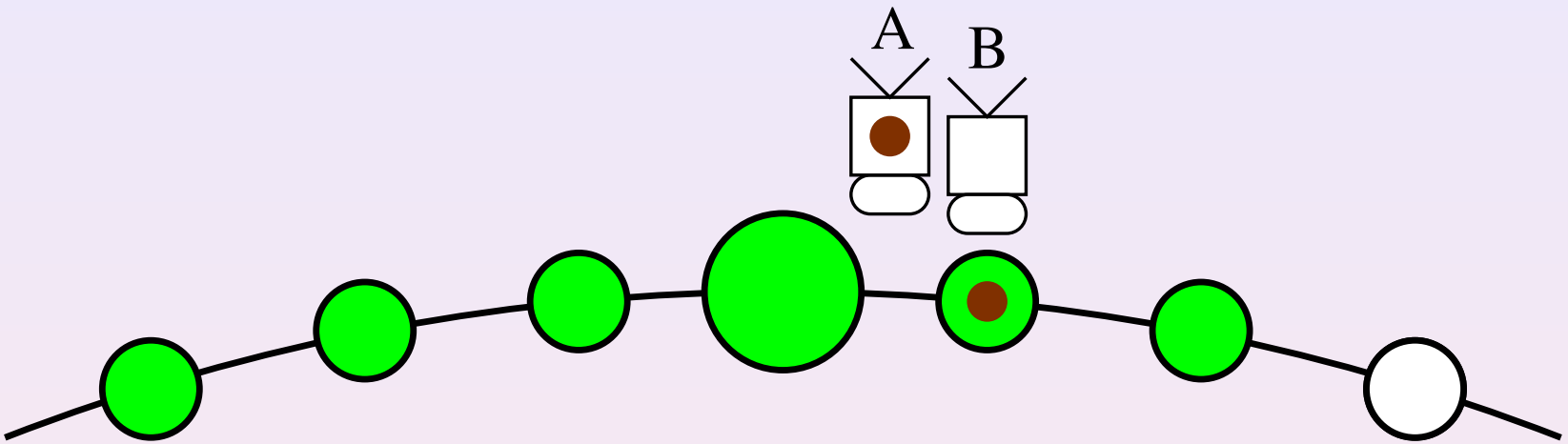


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

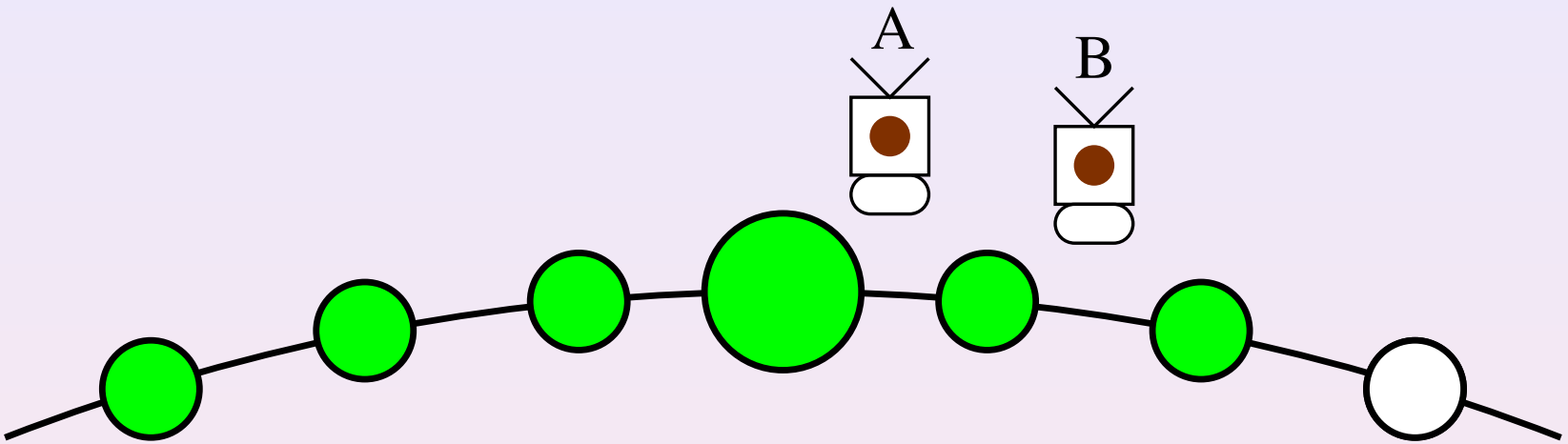


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

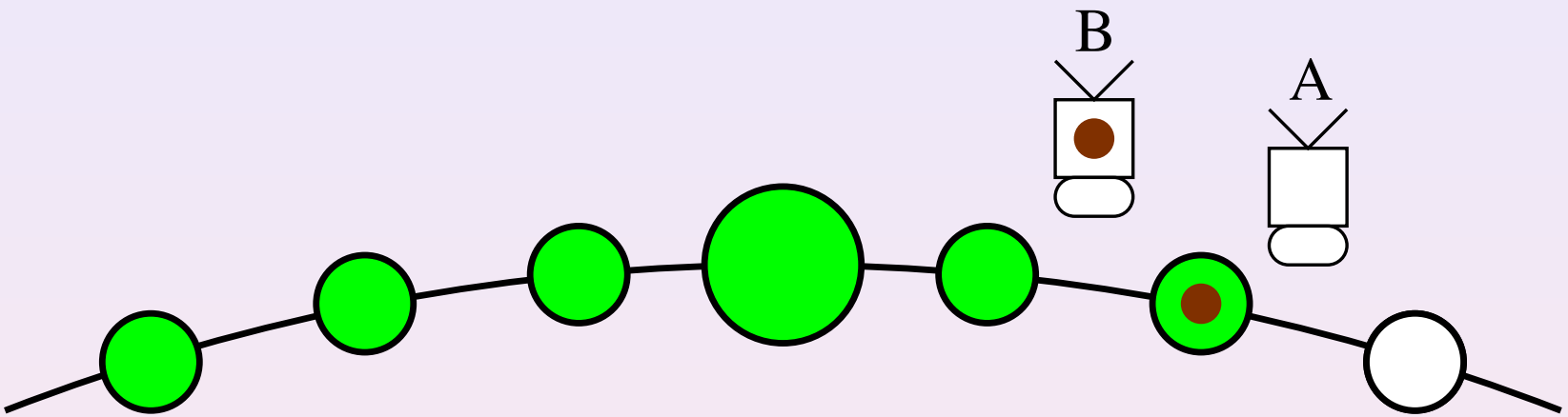


Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong



Déscription rapide

Si l'agent "contrôle"

- 1 caillou → Progression prudente simple vers la droite
- 2 cailloux → Progression prudente double vers la gauche
- 0 caillou → Progression non prudente vers la gauche

Algorithme Ping Pong

$\Rightarrow$ Algorithme correct utilisant $O(n^2)$ déplacements

Déscription rapide

Si l'agent "contrôle"

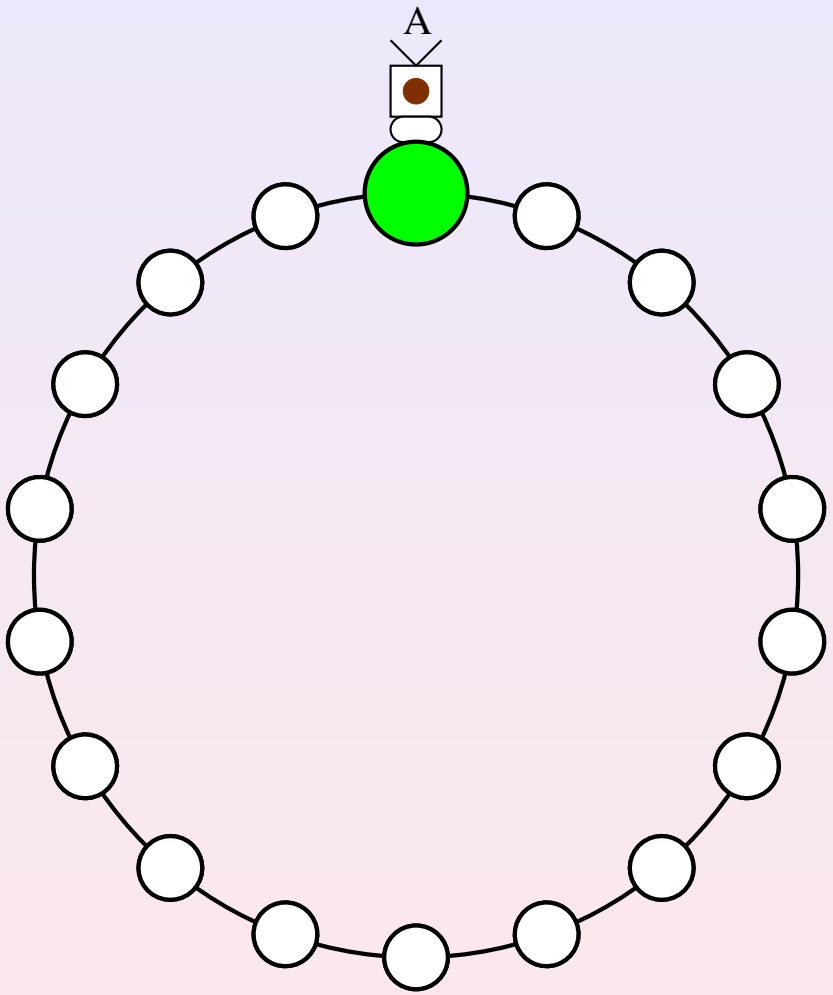
- 1 caillou $\longrightarrow$ Progression prudente simple vers la droite
- 2 cailloux $\longrightarrow$ Progression prudente double vers la gauche
- 0 caillou $\longrightarrow$ Progression non prudente vers la gauche

Algorithme Ping Pong amélioré

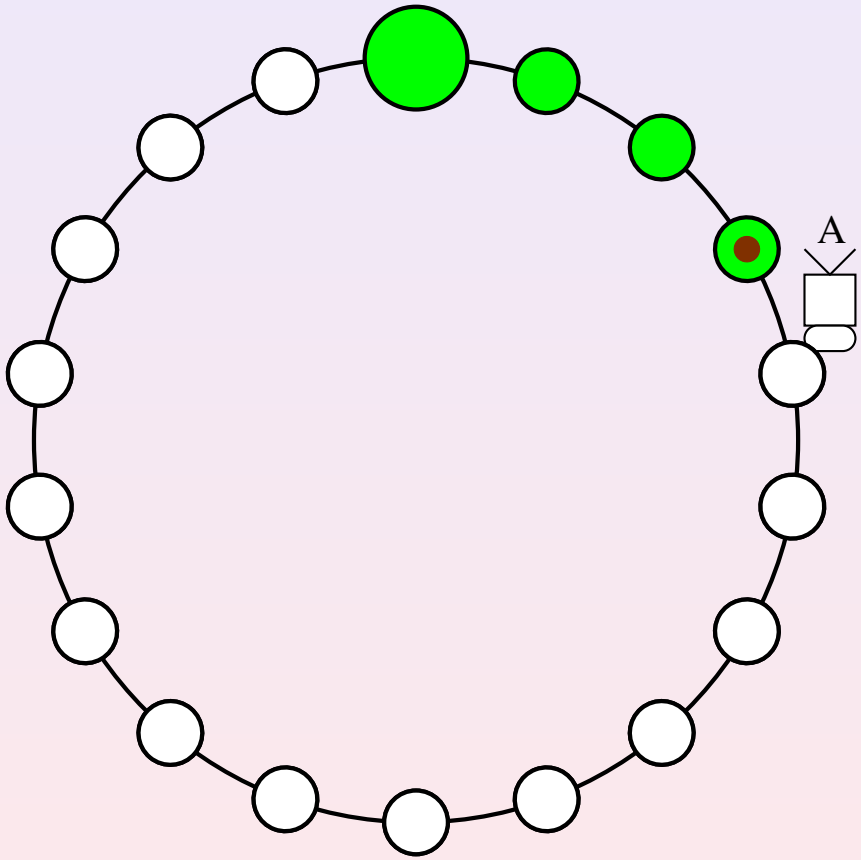
Déscription rapide

- Exécuter Ping Pong **jusqu'à** ce que les agents se "rencontrent" au moins à deux/trois sommets de la base de départ (place saine suffisamment grande)
- **Tant que** plus d'un nœud inexploré
- Diviser la partie inexplorée en deux
- Chaque agent explore une moitié différente
- Le premier agent qui a terminé va aider l'autre à explorer l'autre moitié
- **Lorsqu'**un agent trouve le caillou de l'autre
  - il décale le caillou d'un noeud vers le centre
  - et recommence la boucle

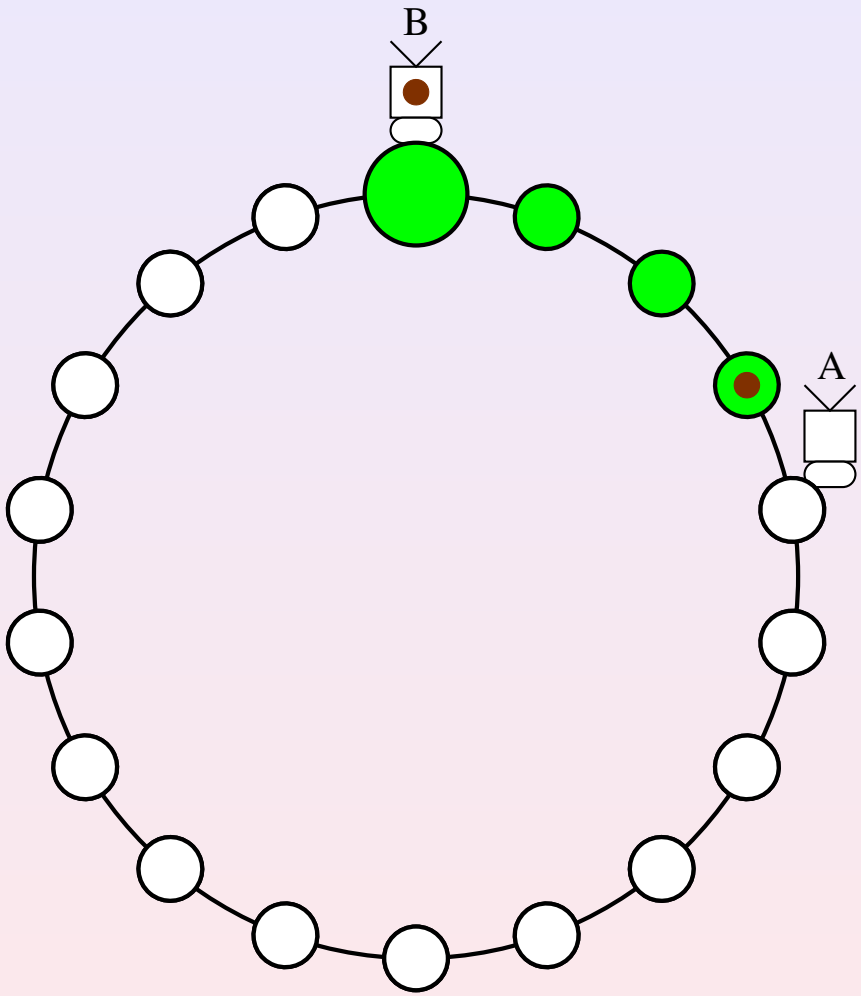
Exemple d'exécution



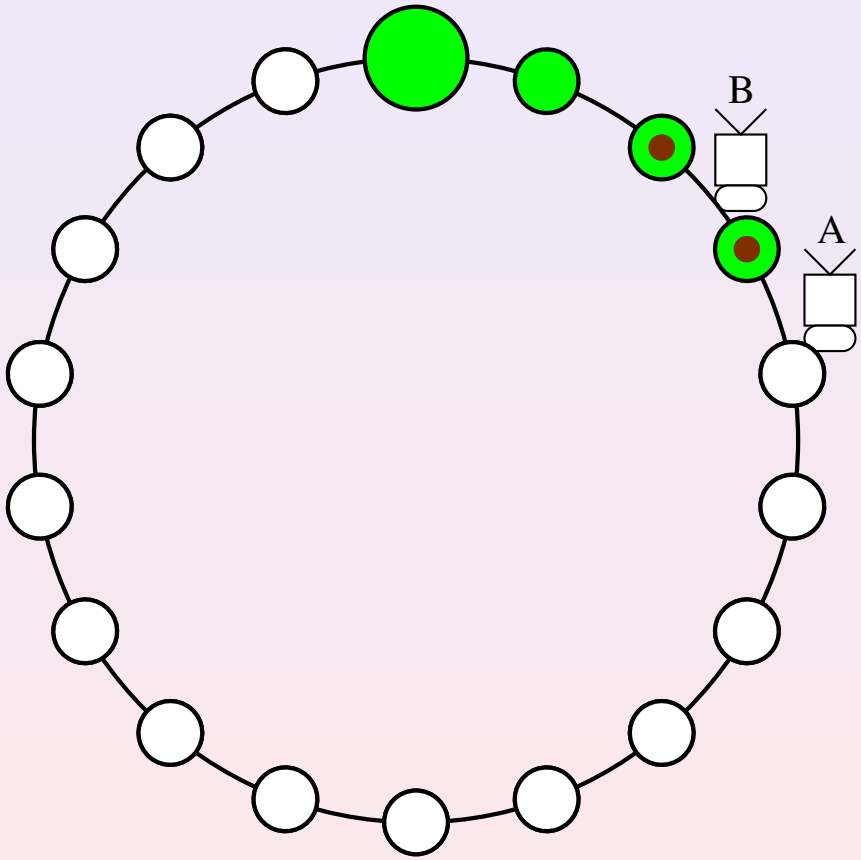
Exemple d'exécution



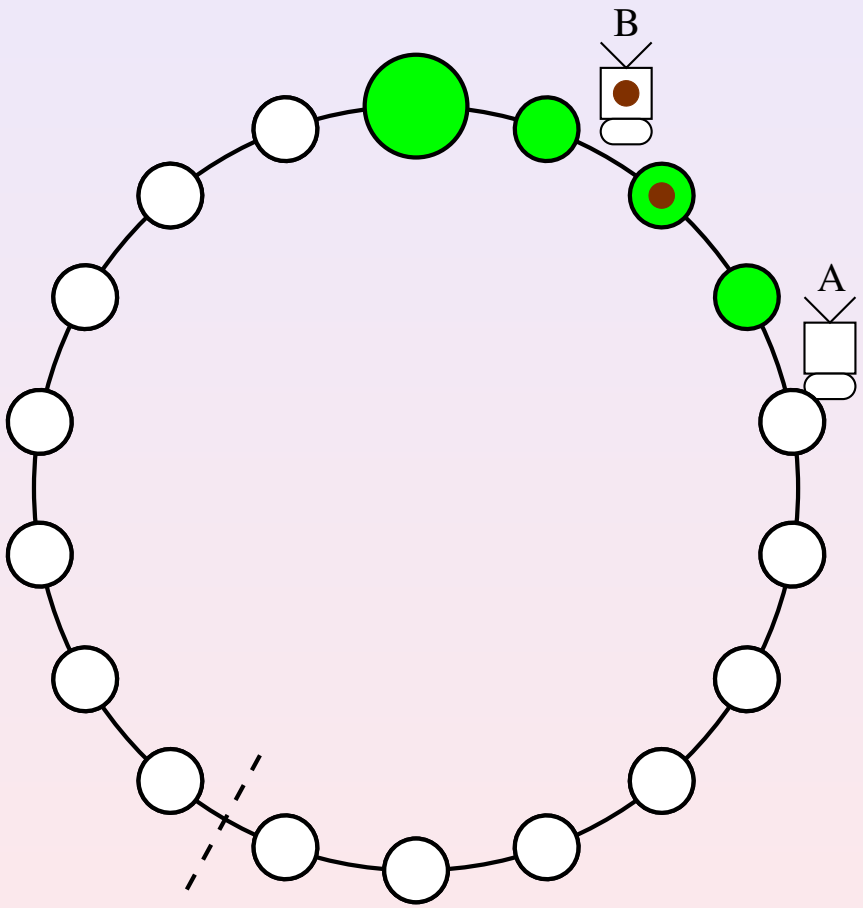
Exemple d'exécution



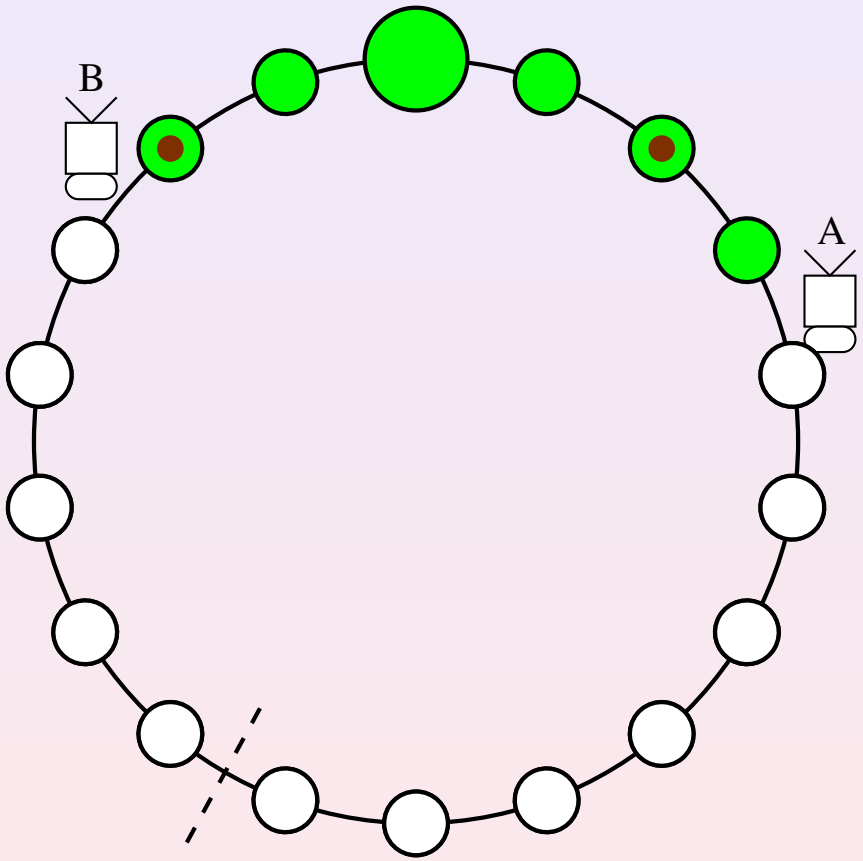
Exemple d'exécution



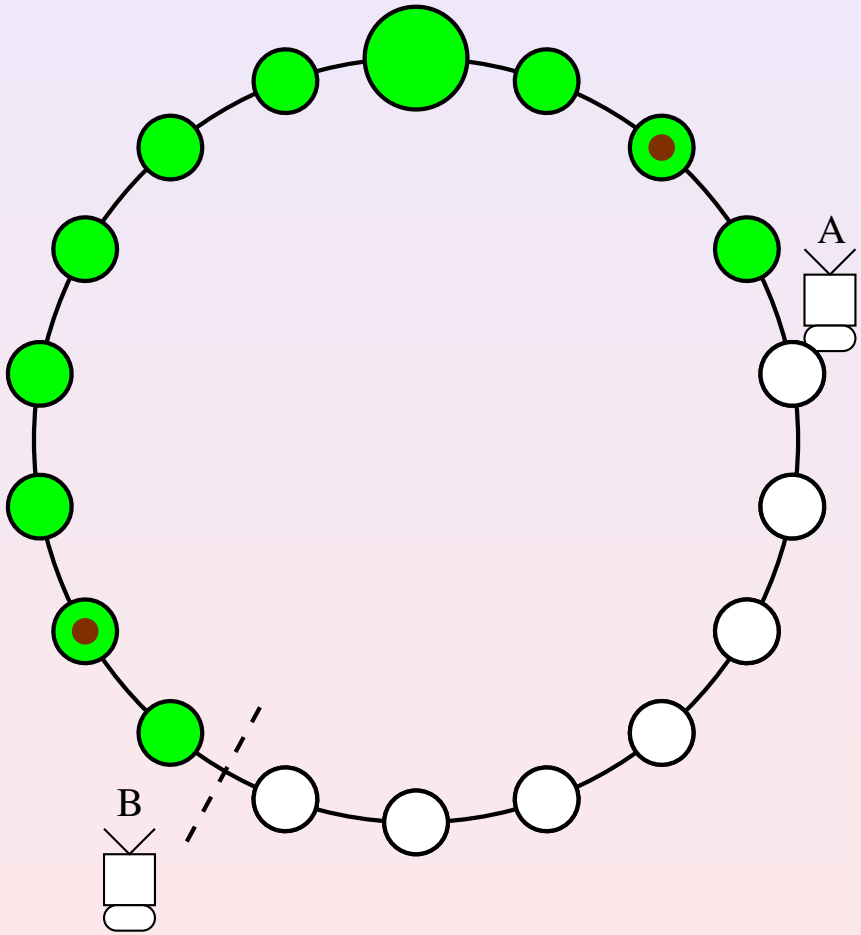
Exemple d'exécution



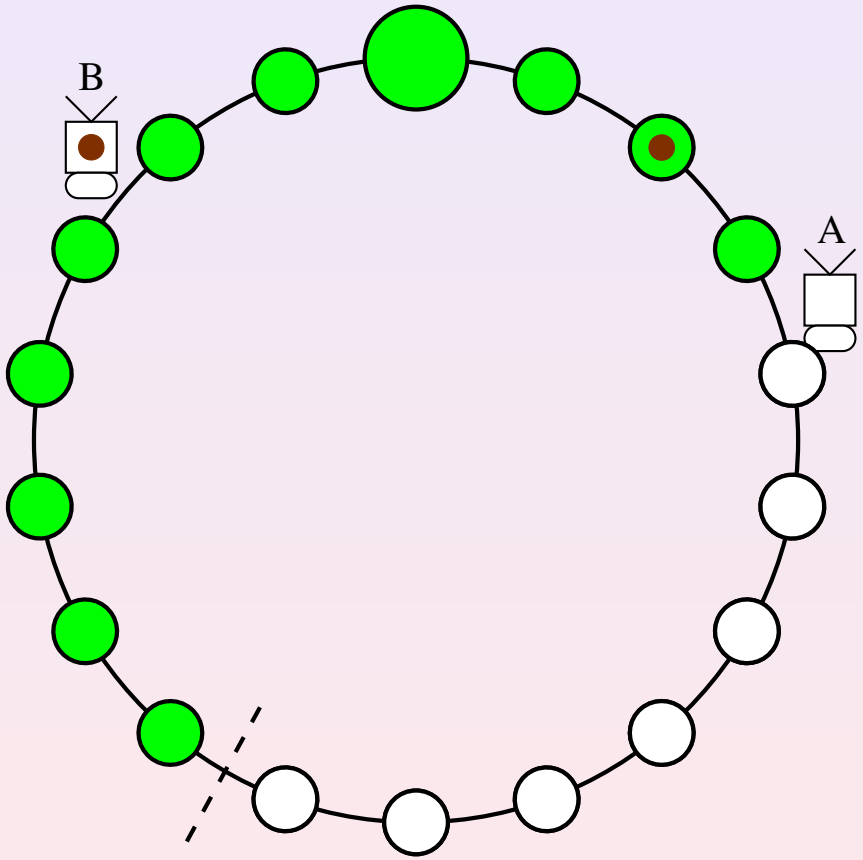
Exemple d'exécution



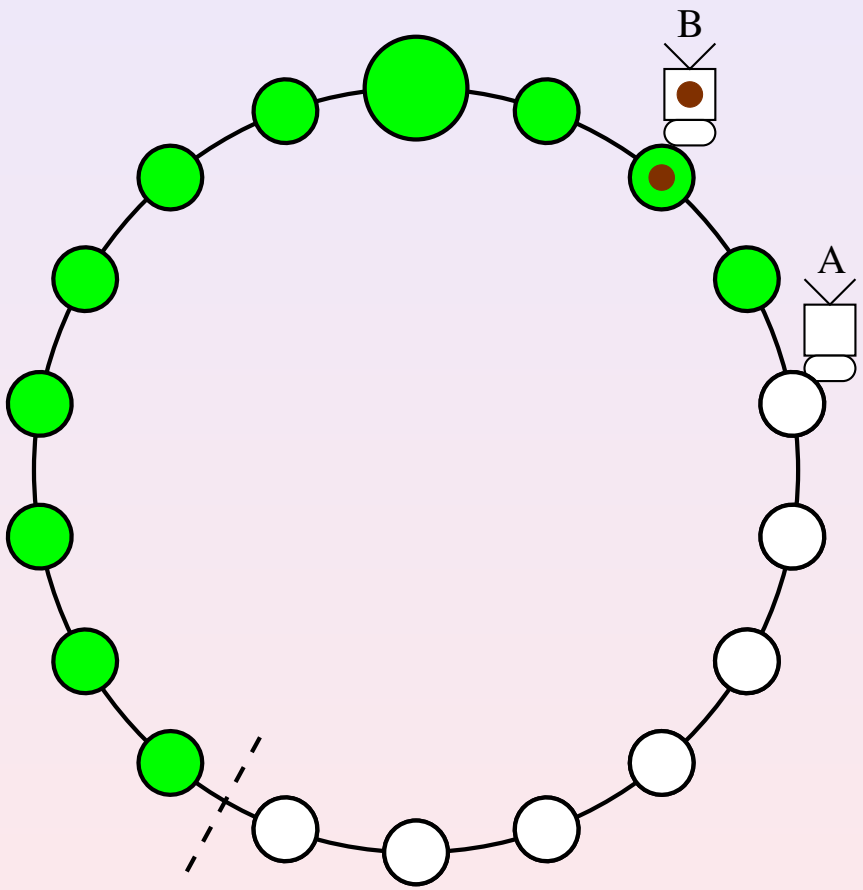
Exemple d'exécution



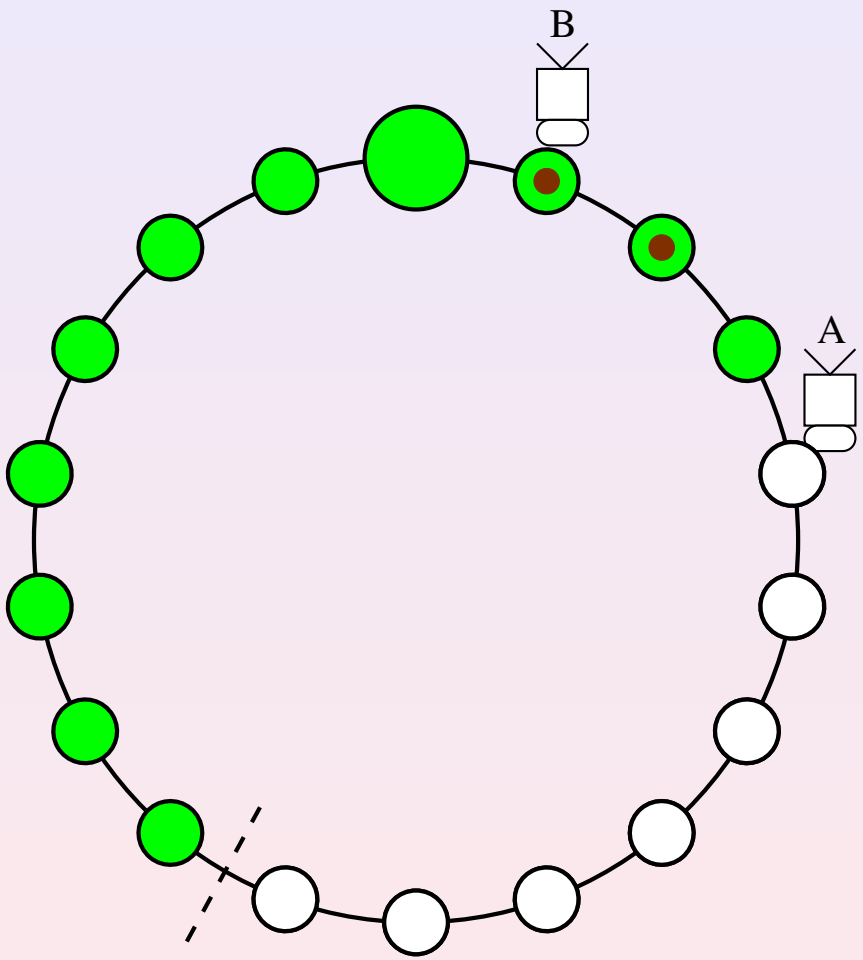
Exemple d'exécution



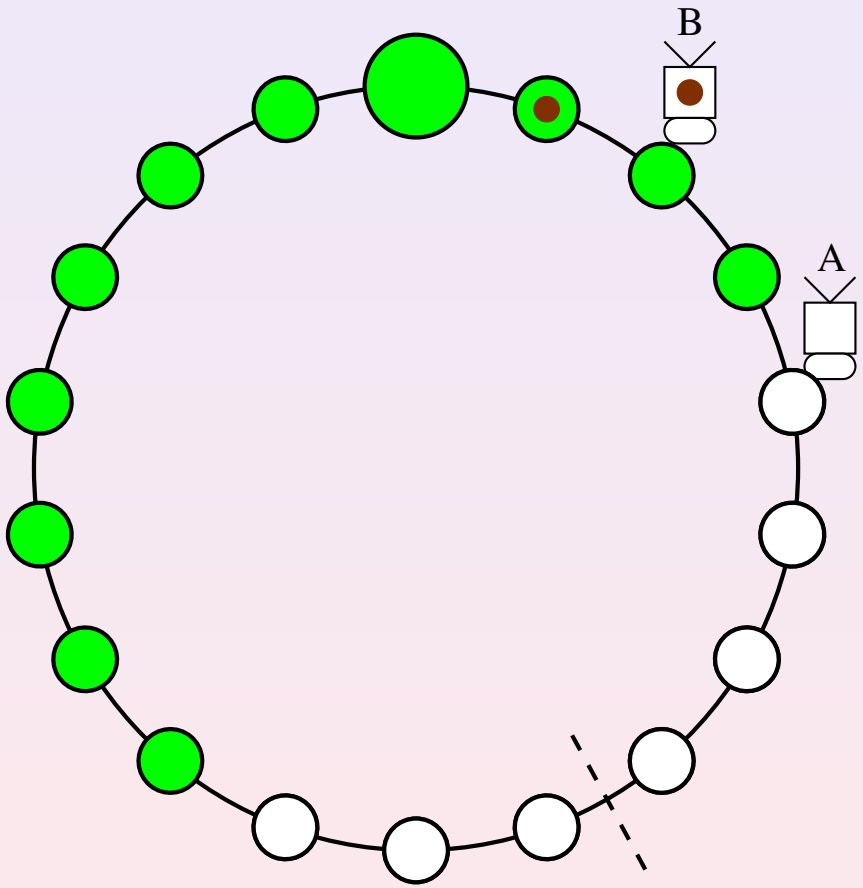
Exemple d'exécution



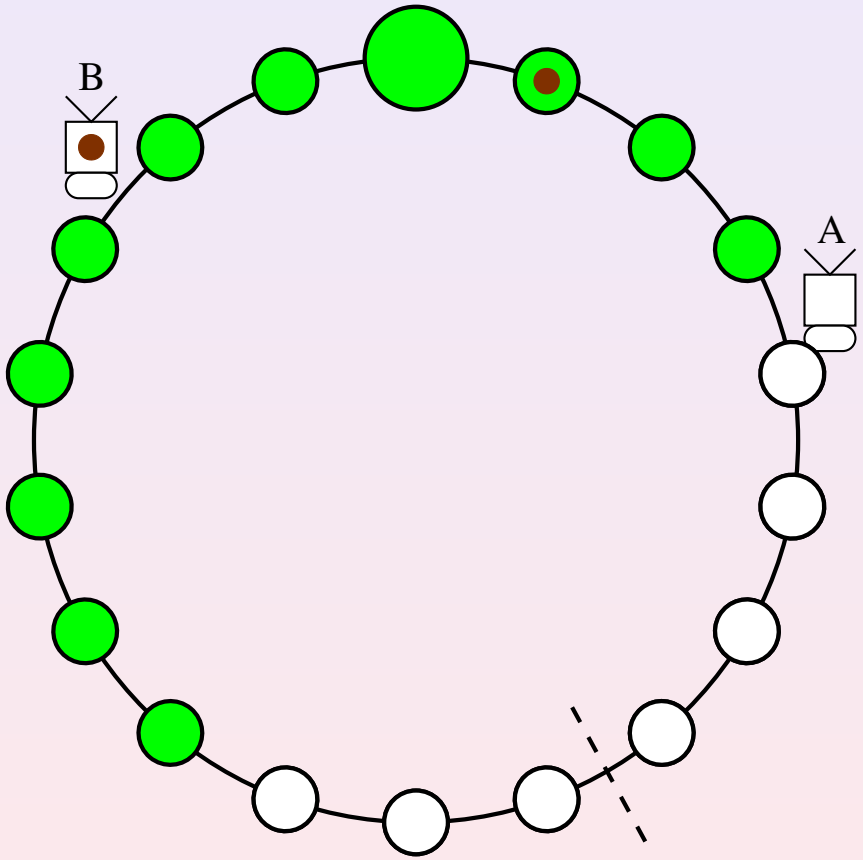
Exemple d'exécution



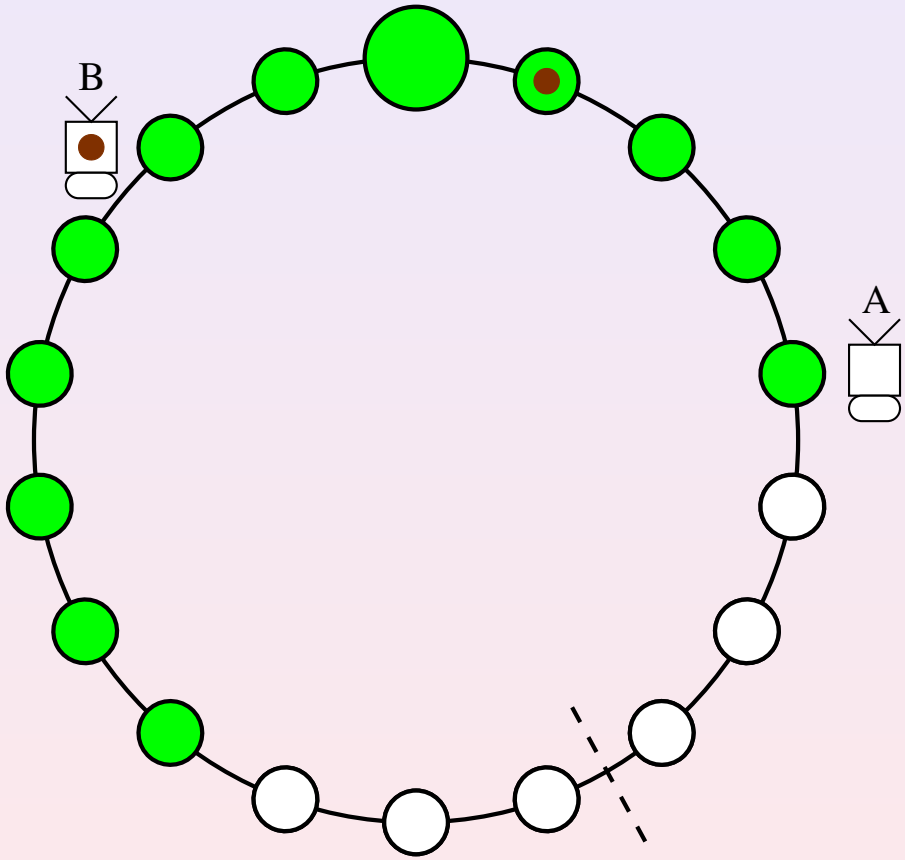
Exemple d'exécution



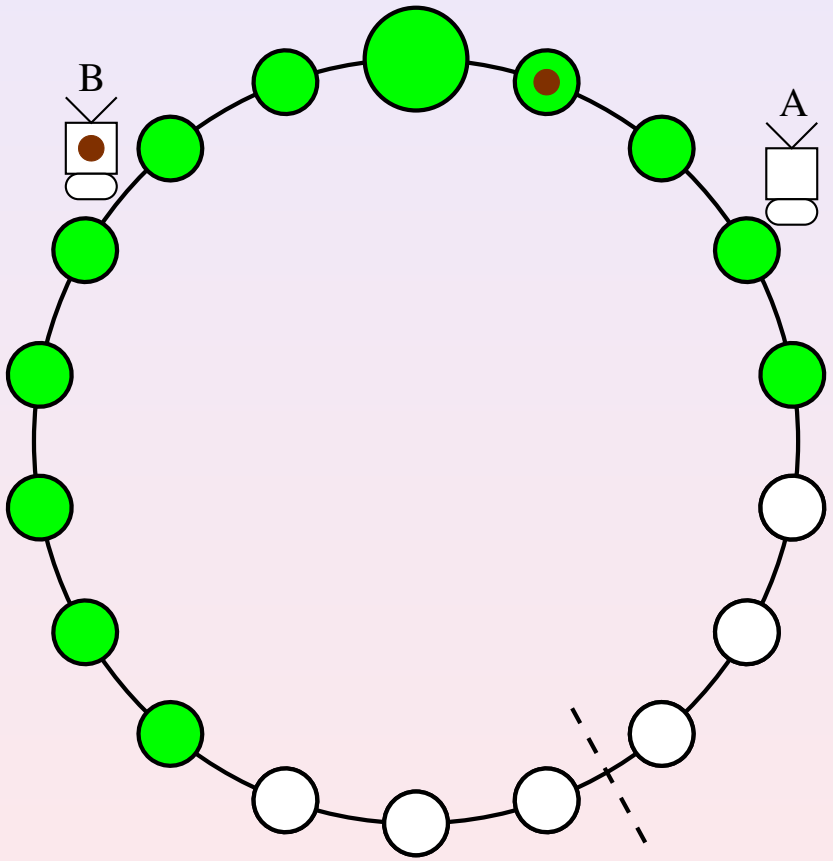
Exemple d'exécution



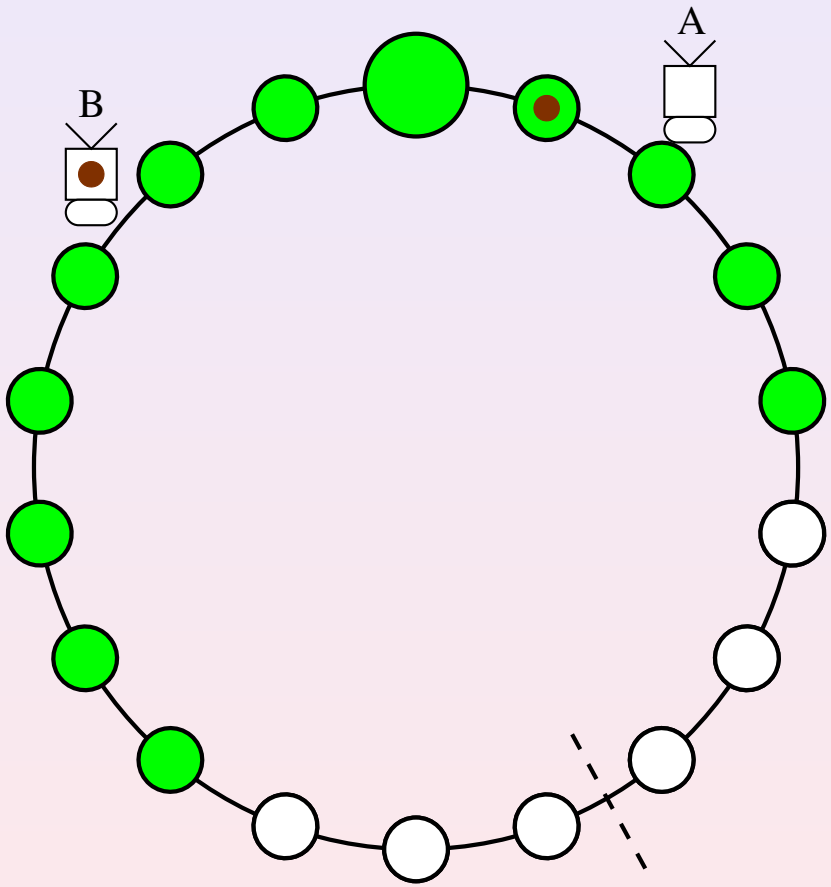
Exemple d'exécution



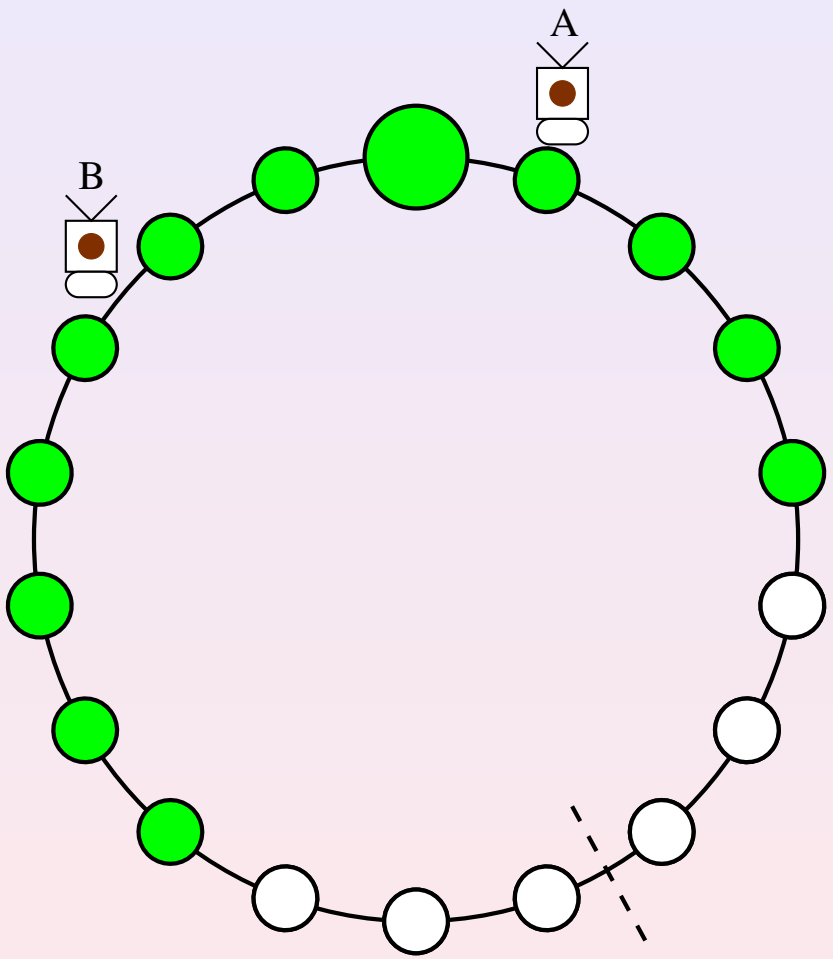
Exemple d'exécution



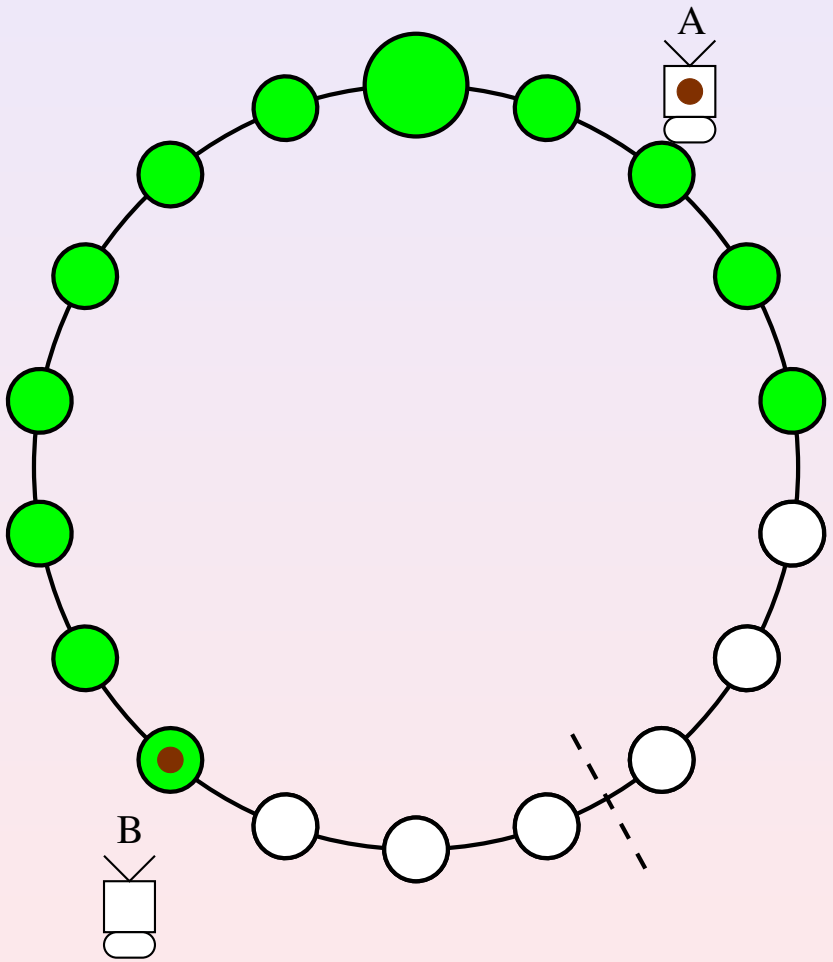
Exemple d'exécution



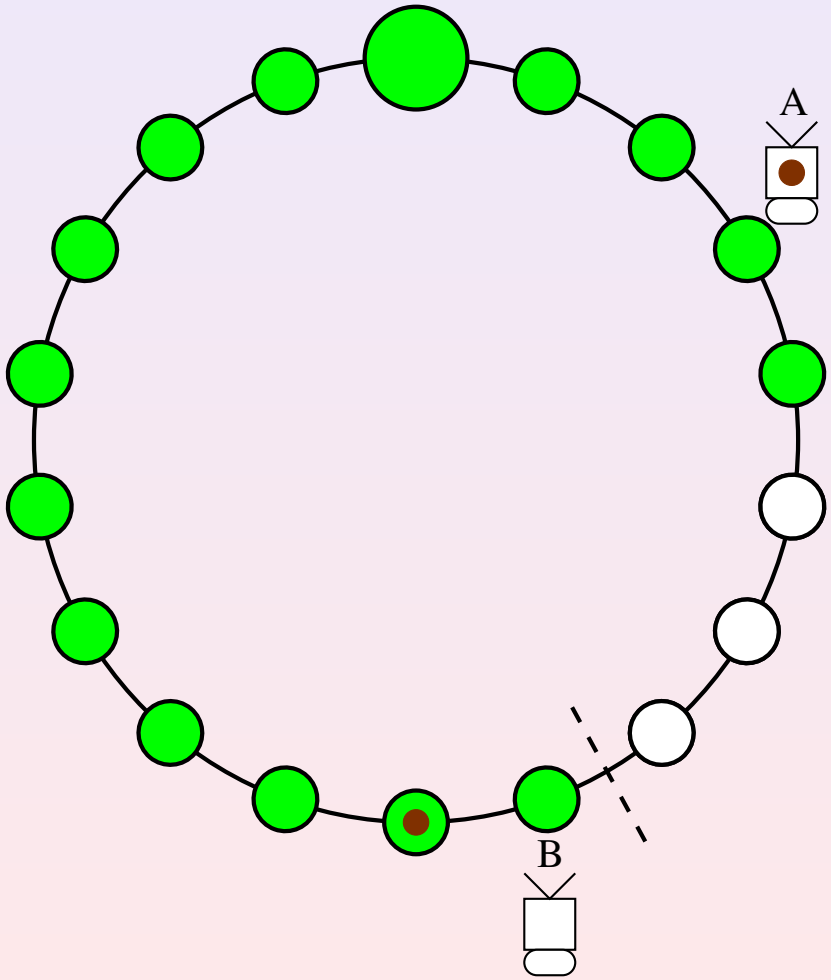
Exemple d'exécution



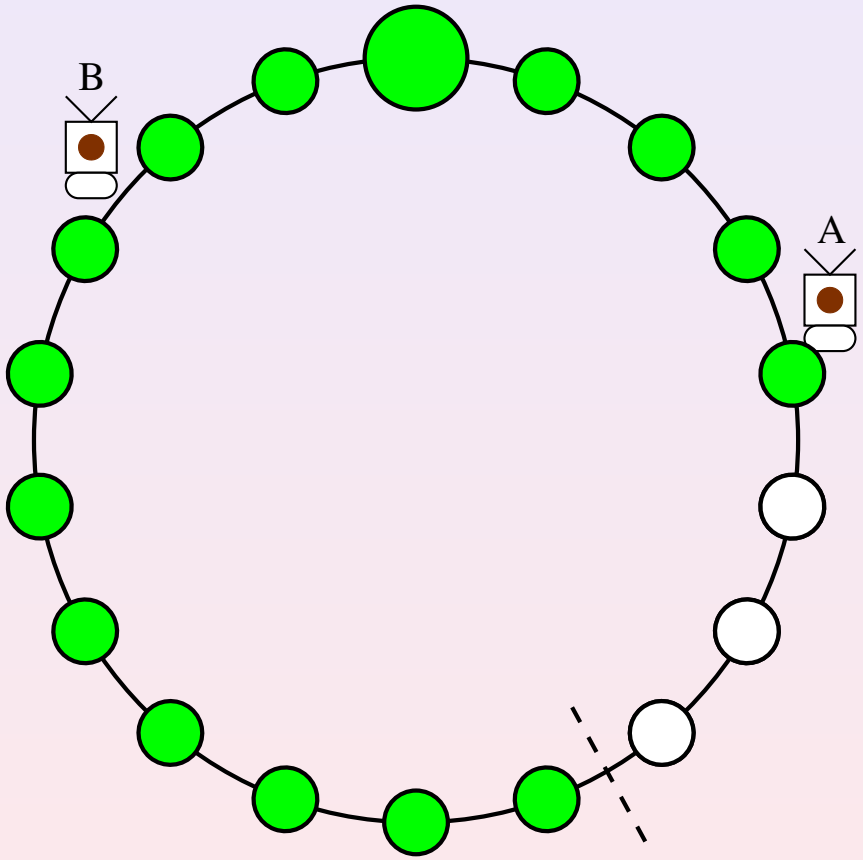
Exemple d'exécution



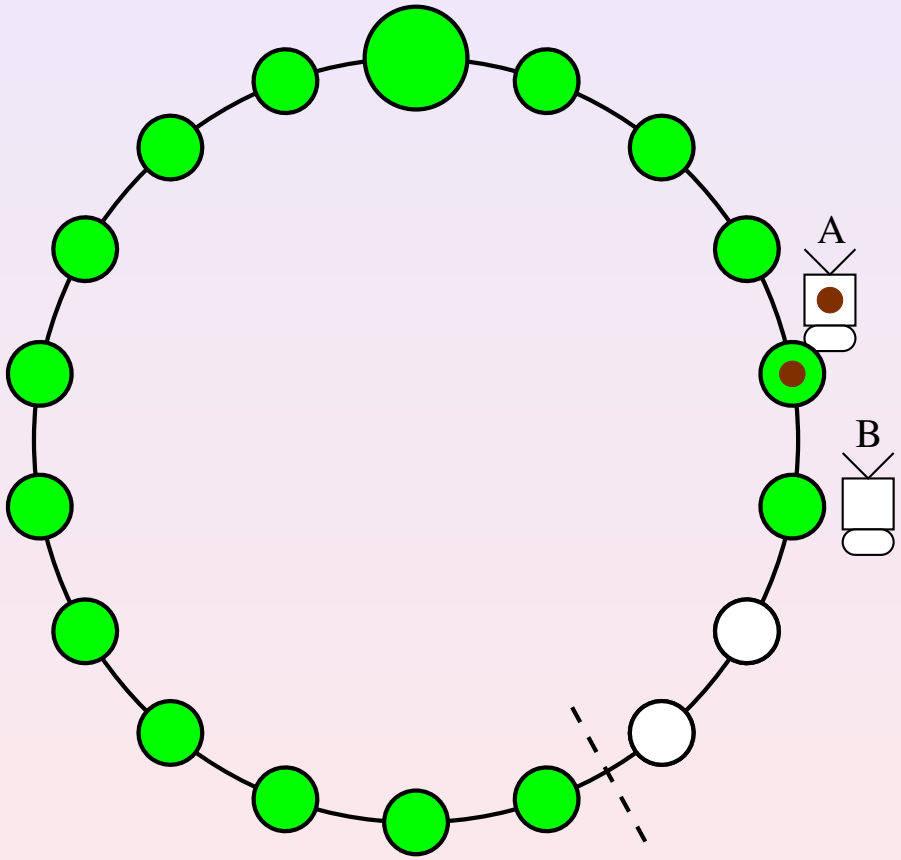
Exemple d'exécution



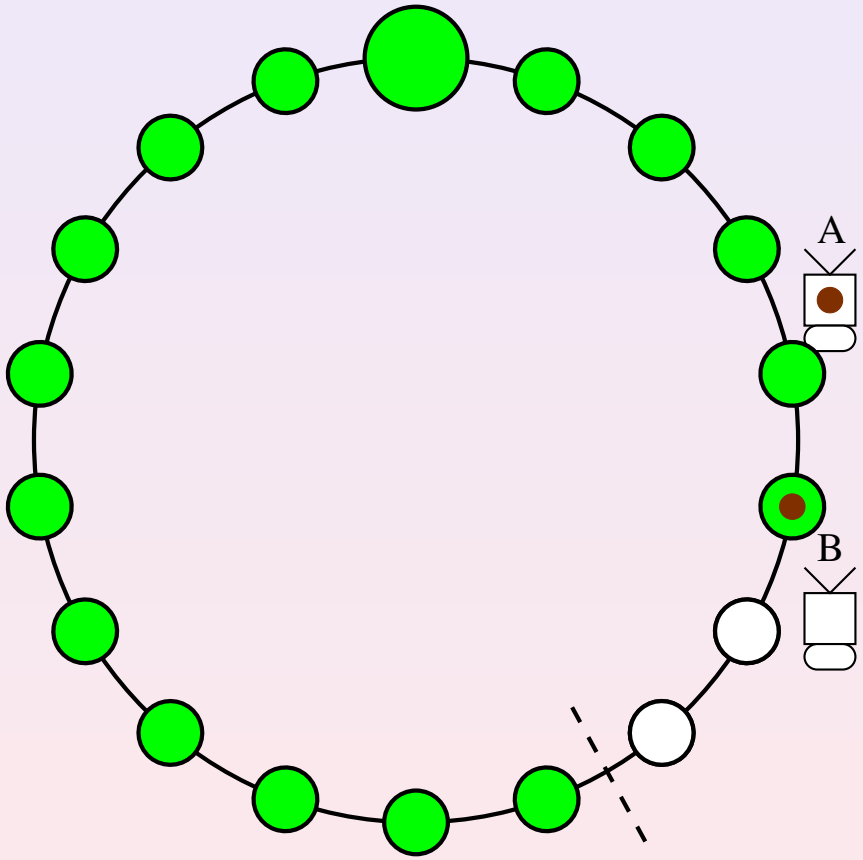
Exemple d'exécution



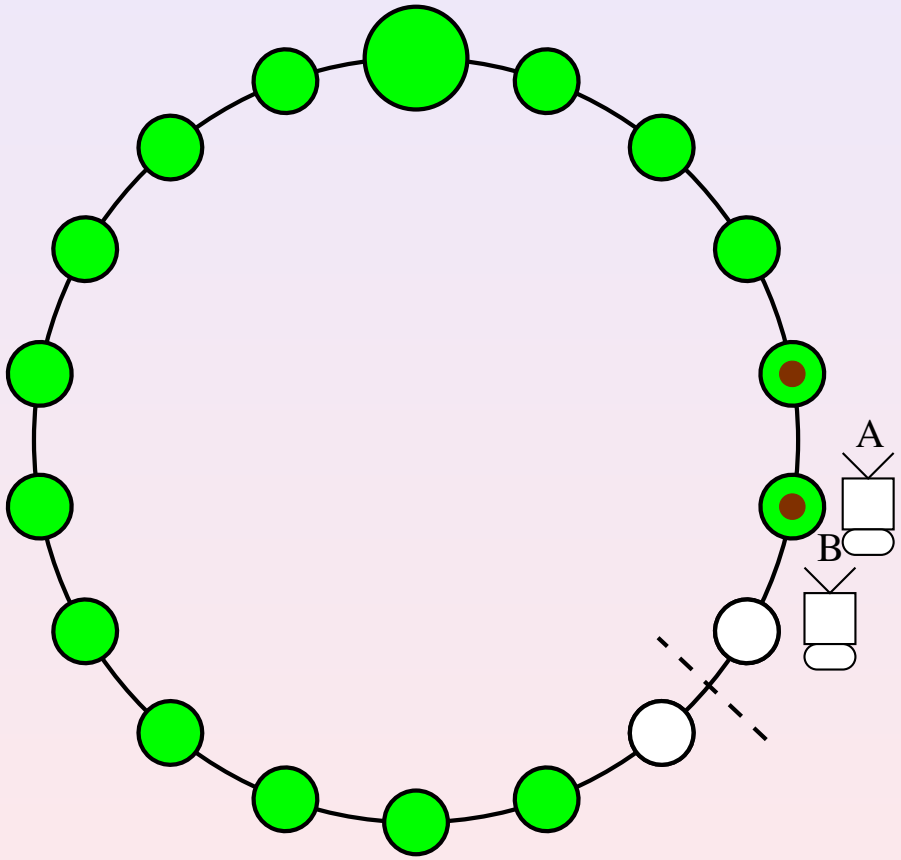
Exemple d'exécution



Exemple d'exécution



Exemple d'exécution



Analyse du Ping Pong amélioré

Analyse

Entre deux divisions/répartitions

- Environ la moitié des nœuds est explorée
- Les agents ont effectués au plus $O(n)$ déplacements

Résultat

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ mouvements

Analyse du Ping Pong amélioré

Analyse

Entre deux divisions/répartitions

- Environ la moitié des nœuds est explorée
- Les agents ont effectués au plus $O(n)$ déplacements

Résultat

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ mouvements

Extension aux graphes arbitraires

Principe de base

Adaptation du Ping Pong amélioré par analogie avec l'anneau

- Ordre d'exploration : DFS d'arbres couvrants (un pour la gauche, un pour la droite)
- i-ème nœud de l'anneau dans une direction $\leftrightarrow$ i-ème nœud exploré dans cette direction
- Arête $\{i, i+1\} \leftrightarrow$ plus court chemin sûr de i à $i+1$

Difficultés

- Une traversée d'arête $\leftrightarrow$ plusieurs traversées d'arêtes
- Changement de paire d'arbres à chaque division

Extension aux graphes arbitraires

Principe de base

Adaptation du Ping Pong amélioré par analogie avec l'anneau

- Ordre d'exploration : DFS d'arbres couvrants (un pour la gauche, un pour la droite)
- i-ème nœud de l'anneau dans une direction $\leftrightarrow$ i-ème nœud exploré dans cette direction
- Arête $\{i, i+1\} \leftrightarrow$ plus court chemin sûr de i à $i+1$

Difficultés

- Une traversée d'arête $\leftrightarrow$ plusieurs traversées d'arêtes
- Changement de paire d'arbres à chaque division

Modification et analyse

Modification

Si possible, utilisation de raccourcis
(aller de i à j sans passer obligatoirement par $i+1, I+2, \dots$)

Analyse

- Longueur moyenne d'une arête de l'anneau constante
- Analyse fine du Ping Pong amélioré : chaque arête est traversée au plus $O(\log n)$ fois

Résultat

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ mouvements

Modification et analyse

Modification

Si possible, utilisation de raccourcis
(aller de i à j sans passer obligatoirement par $i+1, I+2, \dots$)

Analyse

- Longueur moyenne d'une arête de l'anneau constante
- Analyse fine du Ping Pong amélioré : chaque arête est traversée au plus $O(\log n)$ fois

Résultat

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ mouvements

Modification et analyse

Modification

Si possible, utilisation de raccourcis
(aller de i à j sans passer obligatoirement par $i+1, I+2, \dots$)

Analyse

- Longueur moyenne d'une arête de l'anneau constante
- Analyse fine du Ping Pong amélioré : chaque arête est traversée au plus $O(\log n)$ fois

Résultat

- 1 caillou par agent
- 2 agents
- $\Theta(n \log n)$ mouvements

Conclusion et perspectives

Conclusion

Equivalence, pour la recherche de trou noir, entre

- Coordination par tableaux blancs
- Coordination par cailloux

Perspectives

Pour quels autres problèmes d'agents mobiles cette équivalence est-elle vrai ?