Complexité en espace de l'exploration de graphes

David Ilcinkas

Université du Québec en Outaouais, Canada

Séminaire Algorithmique distribuée, LaBRI
8 janvier 2007

Exploration de graphes

Thématique

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

Objectif

Etude de la complexité en espace.

Motivations

- Logique
- Théorie de la complexité
- Robotique et agents logiciels

Exploration de graphes

Thématique

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

Objectif

Etude de la complexité en espace.

Motivations

- Logique
- Théorie de la complexité
- Robotique et agents logiciels

Exploration de graphes

Thématique

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

Objectif

Etude de la complexité en espace.

Motivations

- Logique
- Théorie de la complexité
- Robotique et agents logiciels

Motivations (Logique)

Caractérisation de la classe des langages réguliers

Sur les mots :

- la logique du second ordre monadique
- les automates finis

Sur les arbres :

- la logique du second ordre monadique
- les automates finis de type bottom-up

Langages sur les arbres et sur les graphes

[Engelfriet et Hoogeboom, STACS'06] Equivalence entre :

- Logique du premier ordre avec fermeture transitive
- Automates d'exploration munis de cailloux "imbriqués"

Motivations (Théorie de la complexité)

USTCON (undirected st-connectivity) SL-complet

- $G = \{V, E\}$ graphe non orienté
- $s, t \in V$ deux sommets de G

s et t sont-ils dans la même composante connexe ?

- L = deterministic log-space
- SL ($\supseteq$ L) = symmetric non-deterministic log-space
- NL ($\supseteq$ SL) = non-deterministic log-space

Reingold, STOC 2005
Undirected ST-Connectivity in Log-Space

USTCON $\in$ L $\Rightarrow$ SL=L

Motivations (Robotique et agents logiciels)

Robot physique

- Problème : exploration d'endroits inaccessibles à l'homme
- Mémoire : limitations dues aux contraintes de poids, taille et consommation énergétique

Agent logiciel

- Problème : maintenance d'un réseau informatique
- Mémoire : réseaux de grande taille $\Rightarrow$ impossible de stocker la carte complète

Inconnu, Anonyme

Thématique (rappel)

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

Inconnu

- structure inconnue
- aucune connaissance sur la taille

Anonyme

- sommets sans identifiant
- étiquetage local des arêtes

Inconnu, Anonyme

Thématique (rappel)

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

Inconnu

- structure inconnue
- aucune connaissance sur la taille

Anonyme

- sommets sans identifiant
- étiquetage local des arêtes

Inconnu, Anonyme

Thématique (rappel)

Une entité mobile doit visiter chaque sommet d'un graphe inconnu et anonyme.

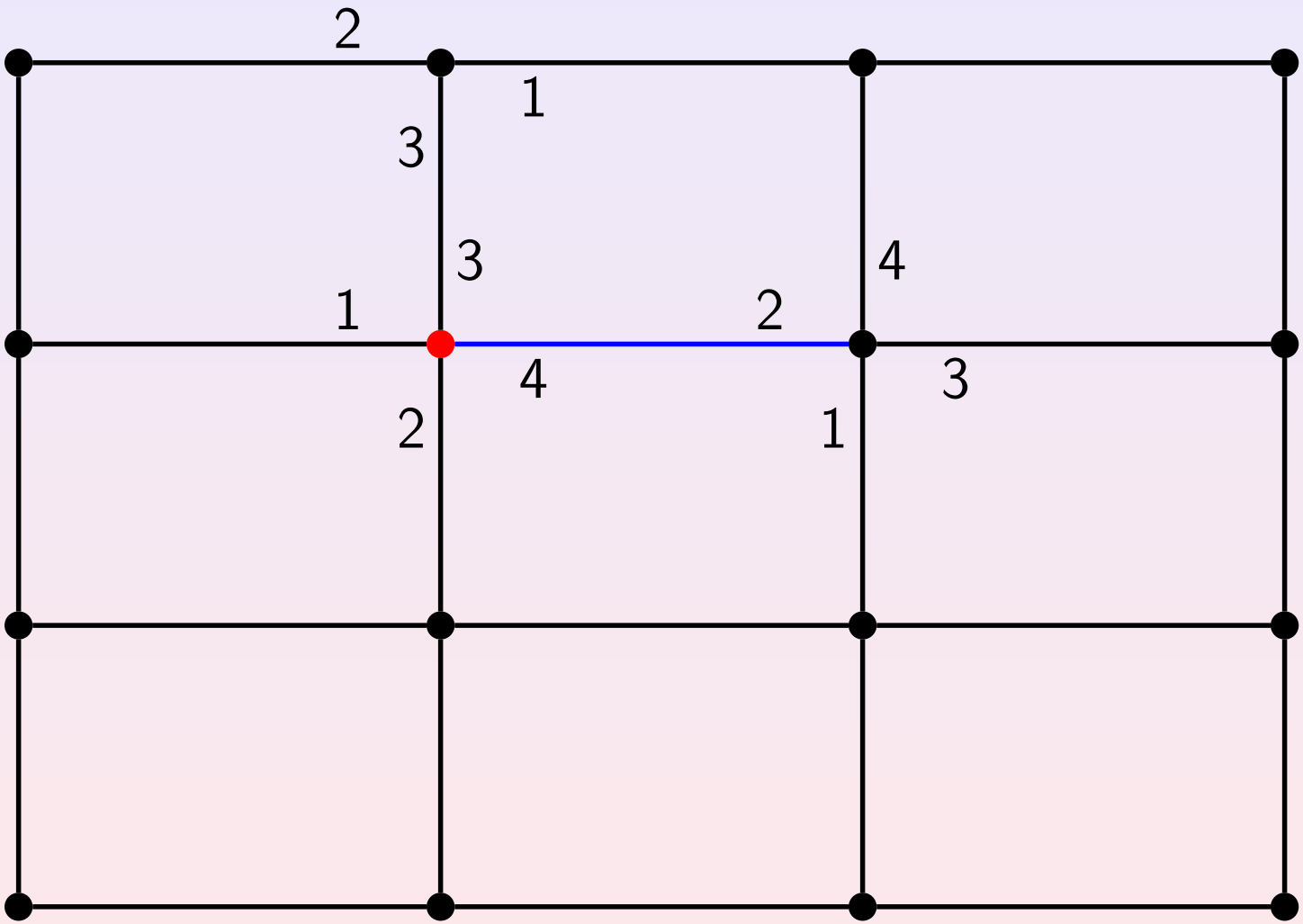
Inconnu

- structure inconnue
- aucune connaissance sur la taille

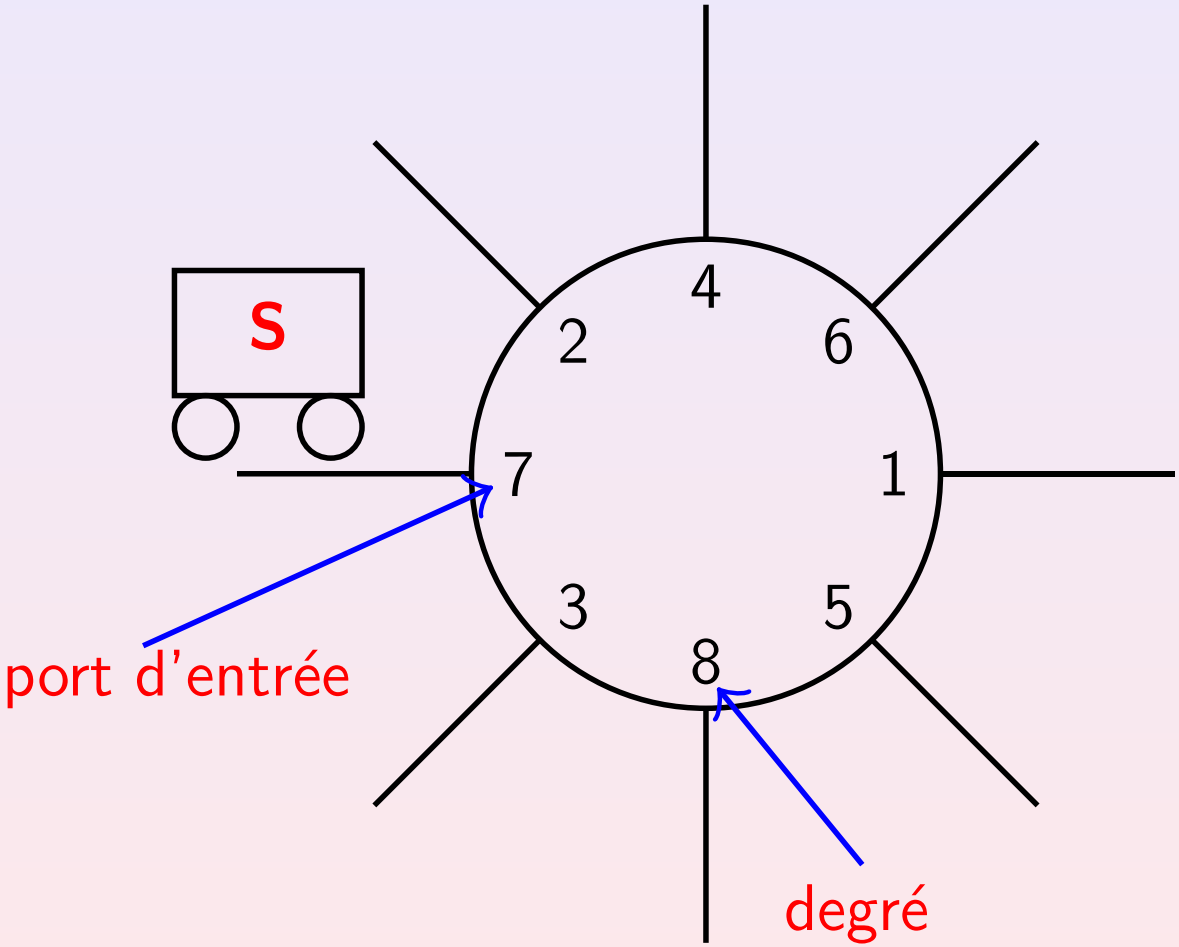
Anonyme

- sommets sans identifiant
- étiquetage local des arêtes

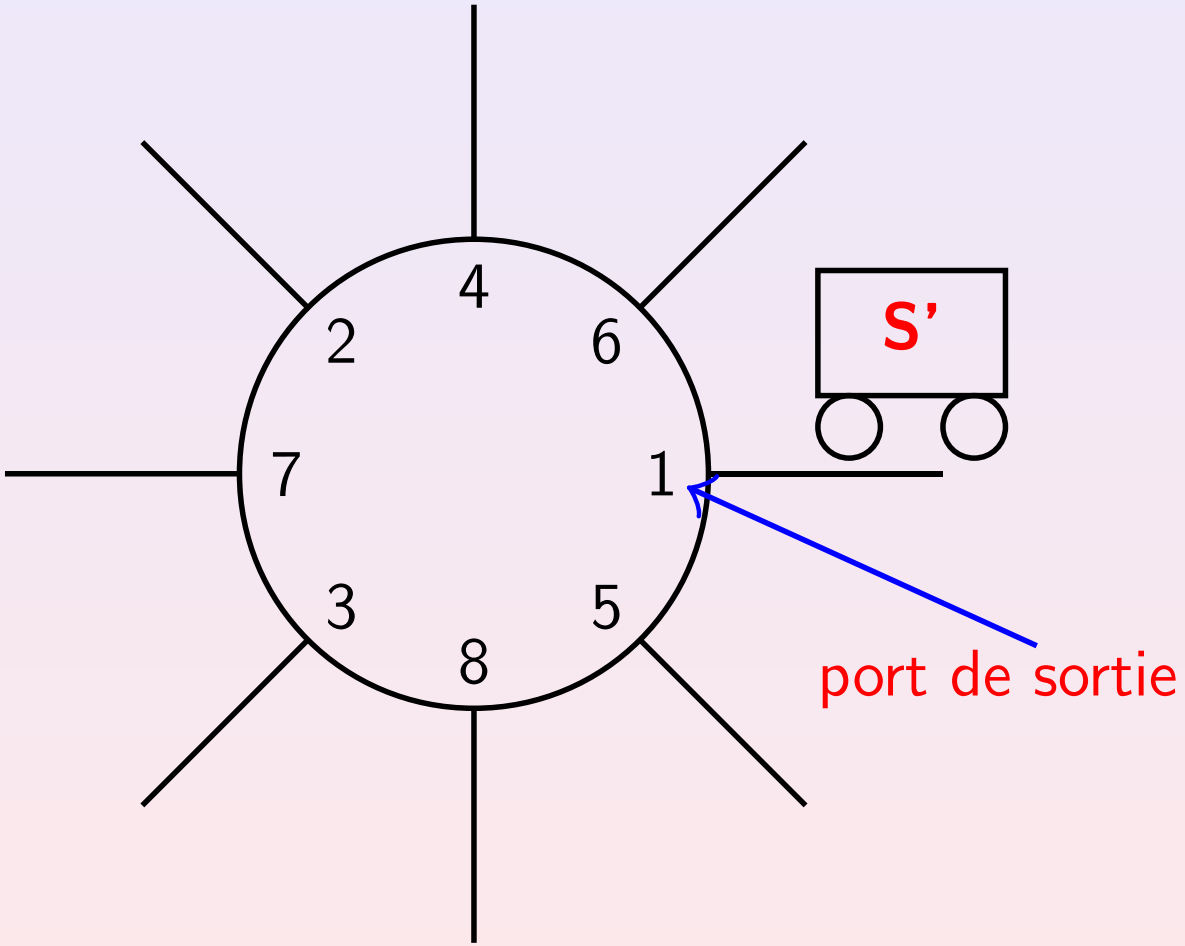
Exemple d'un graphe anonyme



Modèle d'entité mobile : automate



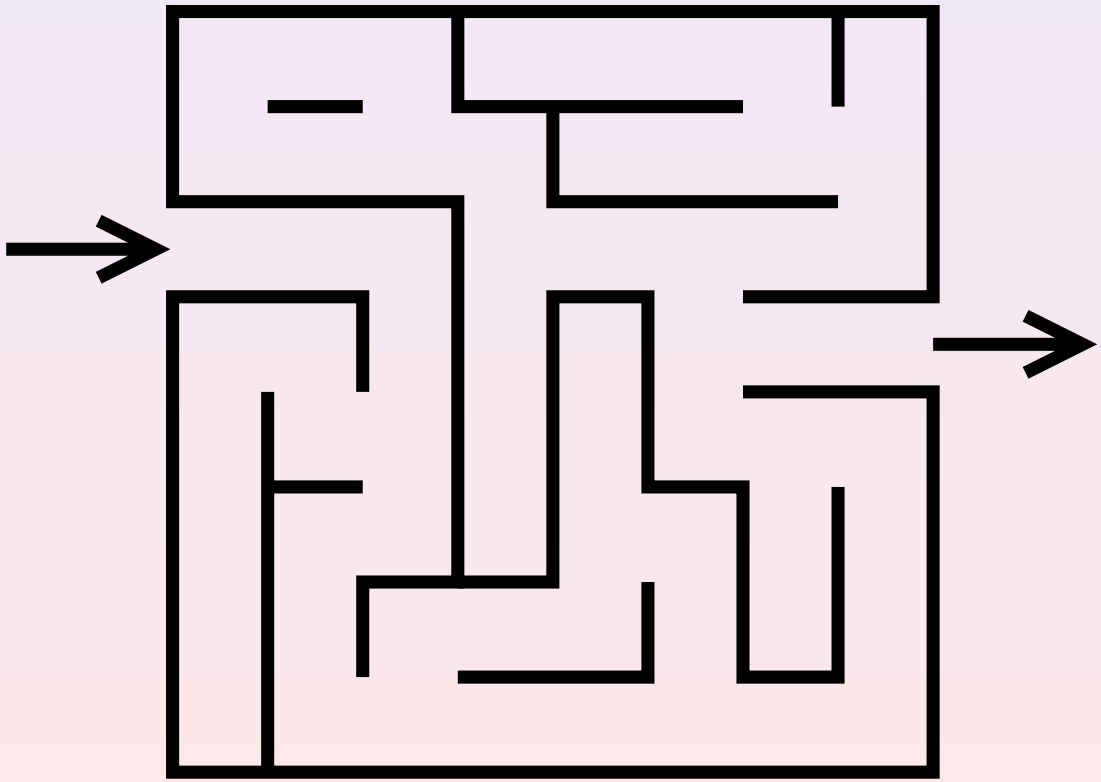
Modèle d'entité mobile : automate



Cas particulier : Labyrinthes

Définition

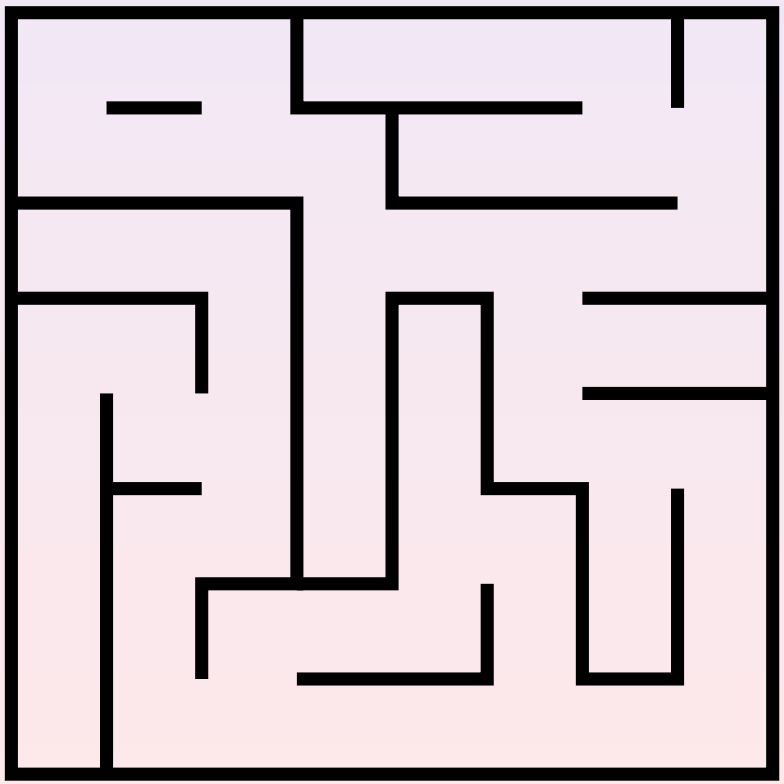
- grille avec des arêtes manquantes
- étiquetage cohérent des arêtes : Nord, Sud, Est, Ouest



Cas particulier : Labyrinthes

Définition

- grille avec des arêtes manquantes
- étiquetage cohérent des arêtes : Nord, Sud, Est, Ouest



Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978
Automata and Labyrinths

$\nexists$ automate fini universel, i.e., explorant tous les labyrinthes

Un caillou est un marqueur pouvant être déposé et repris sur les sommets.

Blum, Kozen, FOCS 1978
On the power of the compass

$\exists$ automate fini universel à deux cailloux pour les labyrinthes
$\exists$ deux automates coopératifs universels pour les labyrinthes
$\nexists$ deux automates coopératifs universels pour les graphes

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978
Automata and Labyrinths

$\nexists$ automate fini universel, i.e., explorant tous les labyrinthes

Un caillou est un marqueur pouvant être déposé et repris sur les sommets.

Blum, Kozen, FOCS 1978
On the power of the compass

$\exists$ automate fini universel à deux cailloux pour les labyrinthes
$\exists$ deux automates coopératifs universels pour les labyrinthes
$\nexists$ deux automates coopératifs universels pour les graphes

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978
Automata and Labyrinths

$\nexists$ automate fini universel, i.e., explorant tous les labyrinthes

Un caillou est un marqueur pouvant être déposé et repris sur les sommets.

Blum, Kozen, FOCS 1978
On the power of the compass

$\exists$ automate fini universel à deux cailloux pour les labyrinthes

$\exists$ deux automates coopératifs universels pour les labyrinthes

$\nexists$ deux automates coopératifs universels pour les graphes

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978
Automata and Labyrinths

$\nexists$ automate fini universel, i.e., explorant tous les labyrinthes

Un caillou est un marqueur pouvant être déposé et repris sur les sommets.

Blum, Kozen, FOCS 1978
On the power of the compass

$\exists$ automate fini universel à deux cailloux pour les labyrinthes
$\exists$ deux automates coopératifs universels pour les labyrinthes
$\nexists$ deux automates coopératifs universels pour les graphes

Résultats sur les labyrinthes

Budach, Math. Nachrichten 1978
Automata and Labyrinths

$\nexists$ automate fini universel, i.e., explorant tous les labyrinthes

Un caillou est un marqueur pouvant être déposé et repris sur les sommets.

Blum, Kozen, FOCS 1978
On the power of the compass

$\exists$ automate fini universel à deux cailloux pour les labyrinthes
$\exists$ deux automates coopératifs universels pour les labyrinthes
$\nexists$ deux automates coopératifs universels pour les graphes

Graphes arbitraires

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Aucune équipe finie d'automates finis n'explore tous les graphes (mêmes planaires et de degré maximum 3).

Un JAG (Jumping Automaton for Graphs) est une équipe d'automates finis coopérant en permanence. Un automate a la possibilité de se téléporter à côté d'un autre automate.

Cook, Rackoff, SIAM J. Comp., 1980
Space lower bounds for maze threadability on restricted machines

Aucun JAG n'explore tous les graphes.

Graphes arbitraires

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Aucune équipe finie d'automates finis n'explore tous les graphes (mêmes planaires et de degré maximum 3).

Un JAG (Jumping Automaton for Graphs) est une équipe d'automates finis coopérant en permanence. Un automate a la possibilité de se téléporter à côté d'un autre automate.

Cook, Rackoff, SIAM J. Comp., 1980
Space lower bounds for maze threadability on restricted machines

Aucun JAG n'explore tous les graphes.

Graphes arbitraires

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Aucune équipe finie d'automates finis n'explore tous les graphes (mêmes planaires et de degré maximum 3).

Un JAG (Jumping Automaton for Graphs) est une équipe d'automates finis coopérant en permanence. Un automate a la possibilité de se téléporter à côté d'un autre automate.

Cook, Rackoff, SIAM J. Comp., 1980
Space lower bounds for maze threadability on restricted machines

Aucun JAG n'explore tous les graphes.

Bornes sur la mémoire

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Pour tout automate à K états, il existe un piège de $O(K)$ sommets.

Corollaire

Un automate explorant tous les graphes de taille au plus n possède au moins $\Omega(n)$ états et donc $\Omega(\log n)$ bits de mémoire.

Reingold, STOC, 2005
Undirected ST-Connectivity in Log-Space

Il existe un automate explorant tous les graphes de taille au plus n possèdant $O(\log n)$ bits de mémoire.

Bornes sur la mémoire

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Pour tout automate à K états, il existe un piège de $O(K)$ sommets.

Corollaire

Un automate explorant tous les graphes de taille au plus n possède au moins $\Omega(n)$ états et donc $\Omega(\log n)$ bits de mémoire.

Reingold, STOC, 2005
Undirected ST-Connectivity in Log-Space

Il existe un automate explorant tous les graphes de taille au plus n possèdant $O(\log n)$ bits de mémoire.

Bornes sur la mémoire

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

Pour tout automate à K états, il existe un piège de $O(K)$ sommets.

Corollaire

Un automate explorant tous les graphes de taille au plus n possède au moins $\Omega(n)$ états et donc $\Omega(\log n)$ bits de mémoire.

Reingold, STOC, 2005
Undirected ST-Connectivity in Log-Space

Il existe un automate explorant tous les graphes de taille au plus n possèdant $O(\log n)$ bits de mémoire.

Graphes orientés

Cook, Rackoff, SIAM J. Comp., 1980

Un JAG a besoin d'une mémoire de $\Omega(\log^2 n / \log \log n)$ bits pour explorer les graphes orientés.

Bender, Slonim, FOCS, 1994

Exploration des graphes orientés par deux robots sans cailloux en temps polynomial.

Fraigniaud, Ilcinkas, Pelc, Peleg, Vahrman, STOC, 1998

- L'exploration des graphes orientés en temps polynomial nécessite $\Omega(\log \log n)$ cailloux.
- Algorithme d'exploration en temps polynomial utilisant $O(\log \log n)$ cailloux.

Graphes orientés

Cook, Rackoff, SIAM J. Comp., 1980

Un JAG a besoin d'une mémoire de $\Omega(\log^2 n / \log\log n)$ bits pour explorer les graphes orientés.

Bender, Slonim, FOCS, 1994

Exploration des graphes orientés par deux robots sans cailloux en temps polynomial.

Bender, Fernandez, Ron, Sahai, Vadhan, STOC, 1998

- L'exploration des graphes orientés en temps polynomial nécessite $\Omega(\log\log n)$ cailloux.
- Algorithme d'exploration en temps polynomial utilisant $O(\log\log n)$ cailloux.

Graphes orientés

Cook, Rackoff, SIAM J. Comp., 1980

Un JAG a besoin d'une mémoire de $\Omega(\log^2 n / \log\log n)$ bits pour explorer les graphes orientés.

Bender, Slonim, FOCS, 1994

Exploration des graphes orientés par deux robots sans cailloux en temps polynomial.

Bender, Fernández, Ron, Sahai, Vadhan, STOC, 1998

- L'exploration des graphes orientés en temps polynomial nécessite $\Omega(\log\log n)$ cailloux.
- Algorithme d'exploration en temps polynomial utilisant $O(\log\log n)$ cailloux.

Plan

Plan

Résultats de la thèse (1)

Graphes non orientés

Exploration classique (perpétuelle)

- Borne inf. : $\Omega(D \log \Delta)$ bits (piège de petit diamètre)
- Borne sup. : $O(D \log \Delta)$ bits (DFS : parcours en profondeur d'abord)

Exploration avec arrêt (automate muni d'un caillou)

- Borne inf. : $\Omega(\log n)$ bits (petit piège)
- Borne sup. : $O(D \log \Delta)$ bits (DFS contrôlé)

Hiérarchie des automates

$\mathcal{G}_k = \{\text{graphes explorables par un automate de } k \text{ bits}\}$

$\forall k, \mathcal{G}_k \neq \mathcal{G}_{10k}$

Résultats de la thèse (1)

Graphes non orientés

Exploration classique (perpétuelle)

- Borne inf. : $\Omega(D \log \Delta)$ bits (piège de petit diamètre)
- Borne sup. : $O(D \log \Delta)$ bits (DFS : parcours en profondeur d'abord)

Exploration avec arrêt (automate muni d'un caillou)

- Borne inf. : $\Omega(\log n)$ bits (petit piège)
- Borne sup. : $O(D \log \Delta)$ bits (DFS contrôlé)

Hiérarchie des automates

$\mathcal{G}_k = \{\text{graphes explorables par un automate de } k \text{ bits}\}$

$$\forall k, \mathcal{G}_k \neq \mathcal{G}_{10k}$$

Résultats de la thèse (2)

Graphes orientés

Exploration avec arrêt (automate muni de cailloux)

- Borne inf. : $\Omega(n \log \Delta)$ bits
- Borne sup. : $O(n\Delta \log n)$ bits (taille d'une carte)
  (temps exponentiel, 1 caillou)
- Borne sup. : $O(n^2\Delta \log n)$ bits
  (temps polynomial, $\Theta(\log \log n)$ cailloux)

Résultats (Fraigniaud, I., Rajsbaum, Tixeuil)

Théorème

Pour tout automate à K états muni d'un caillou, il existe un piège planaire de degré maximum 3 de taille $O(K^3)$.

Corollaire

Un automate réalisant l'exploration avec arrêt dans tous les graphes d'au plus n sommets possède au minimum $\Omega(\log n)$ bits de mémoire.

Résultats (Fraigniaud, I., Rajsbaum, Tixeuil)

Théorème

Pour tout automate à K états muni d'un caillou, il existe un piège planaire de degré maximum 3 de taille $O(K^3)$.

Corollaire

Un automate réalisant l'exploration avec arrêt dans tous les graphes d'au plus n sommets possède au minimum $\Omega(\log n)$ bits de mémoire.

Automate de Moore

Entrées

- S : état courant
- i : port d'entrée
- d : degré du nœud

Fonction de transition

nouvel état : $S' = f(S, i, d)$

Fonction de sortie

port de sortie : $j = \lambda(S')$

Automate de Moore

Entrées

- S : état courant
- i : port d'entrée
- d : degré du nœud

Fonction de transition

nouvel état : $S' = f(S, i, d)$

Fonction de sortie

port de sortie : $j = \lambda(S')$

Automate de Moore

Entrées

- S : état courant
- i : port d'entrée
- d : degré du nœud

Fonction de transition

nouvel état : $S' = f(S, i, d)$

Fonction de sortie

port de sortie : $j = \lambda(S')$

Graphes homogènes

Graphe d-homogène

- d-régulier
- à arêtes colorées (même numéro de port aux deux extrémités)

Dans les graphes d-homogènes :

- $S' = f(S, i, d)$ (automate de Moore)
- $S' = f(S, \lambda(S), d)$ (arêtes colorées)
- $S' = g(S)$ (d-régulier)

Graphes homogènes

Graphe d-homogène

- d-régulier
- à arêtes colorées (même numéro de port aux deux extrémités)

Dans les graphes d-homogènes :

- $S' = f(S, i, d)$ (automate de Moore)
- $S' = f(S, \lambda(S), d)$ (arêtes colorées)
- $S' = g(S)$ (d-régulier)

Graphes homogènes

Graphe d-homogène

- d-régulier
- à arêtes colorées (même numéro de port aux deux extrémités)

Dans les graphes d-homogènes :

- $S' = f(S, i, d)$ (automate de Moore)
- $S' = f(S, \lambda(S), d)$ (arêtes colorées)
- $S' = g(S)$ (d-régulier)

Graphes homogènes

Graphe d-homogène

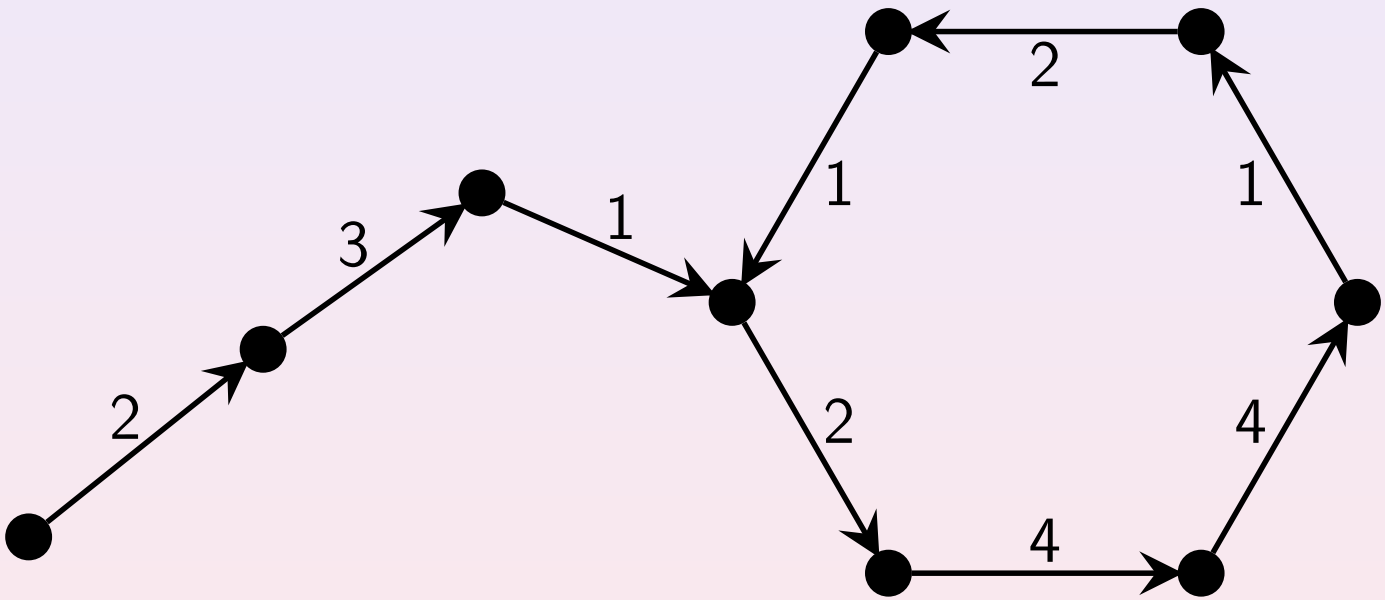
- d-régulier
- à arêtes colorées (même numéro de port aux deux extrémités)

Dans les graphes d-homogènes :

- $S' = f(S, i, d)$ (automate de Moore)
- $S' = f(S, \lambda(S), d)$ (arêtes colorées)
- $S' = g(S)$ (d-régulier)

Automate réduit (1)

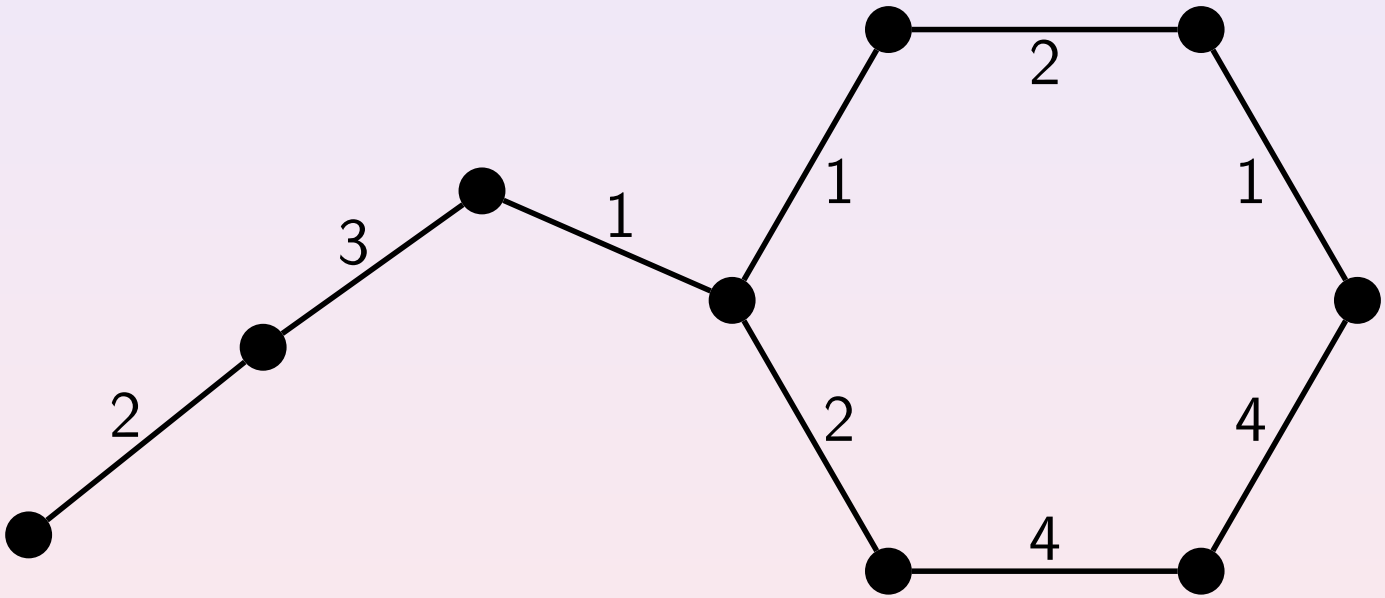
Dans les graphes d-homogènes, le graphe de transition de l'automate est un 1-facteur.



Empreinte de l'automate : $231(244121)^\infty$

Automate réduit (1)

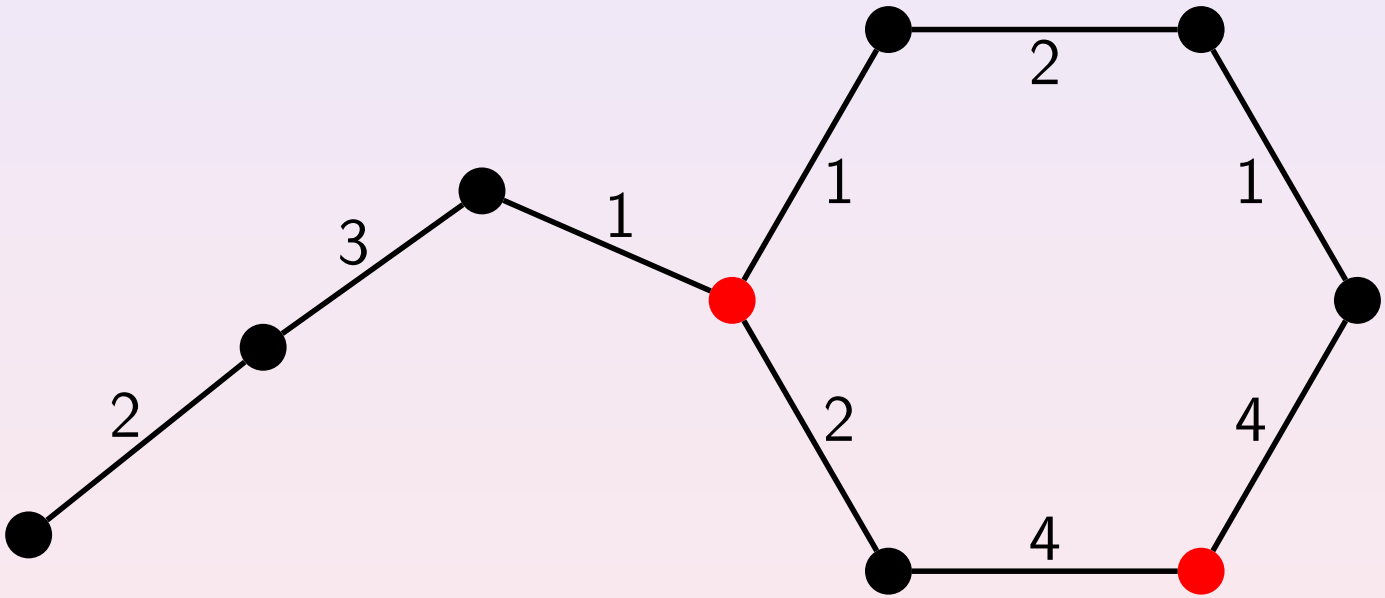
Dans les graphes d-homogènes, le graphe de transition de l'automate est un 1-facteur.



Empreinte de l'automate : $231(244121)^\infty$

Automate réduit (1)

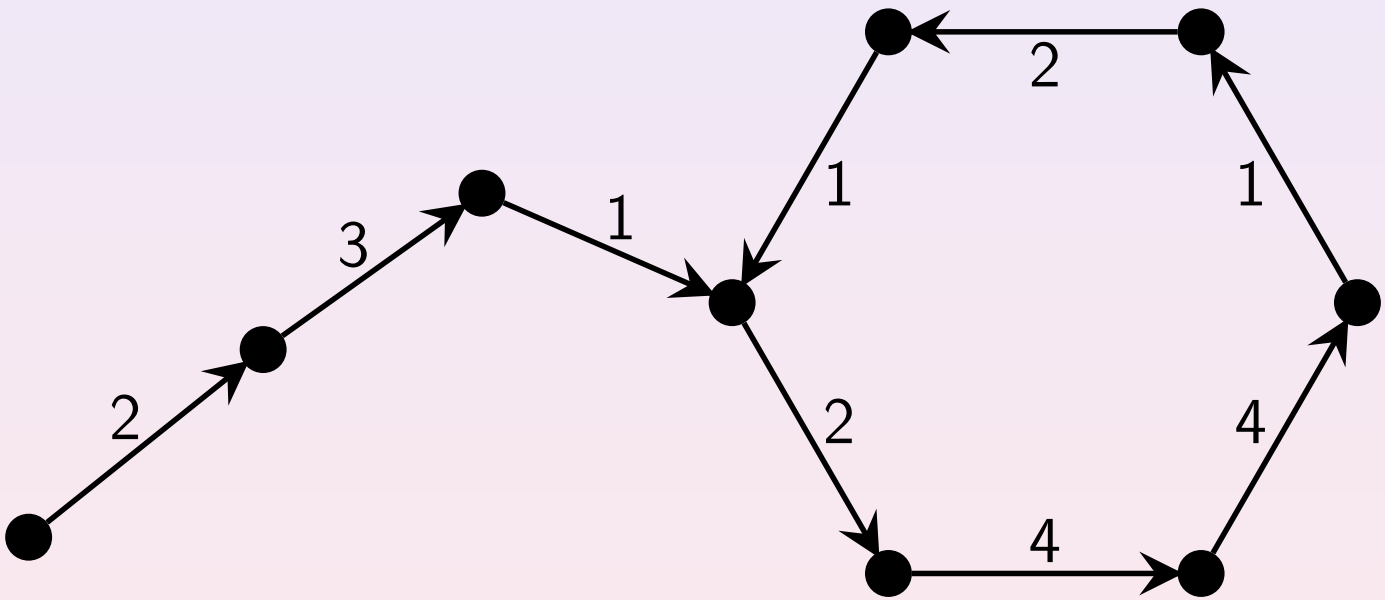
Dans les graphes d-homogènes, le graphe de transition de l'automate est un 1-facteur.



Empreinte de l'automate : $231(244121)^\infty$

Automate réduit (2)

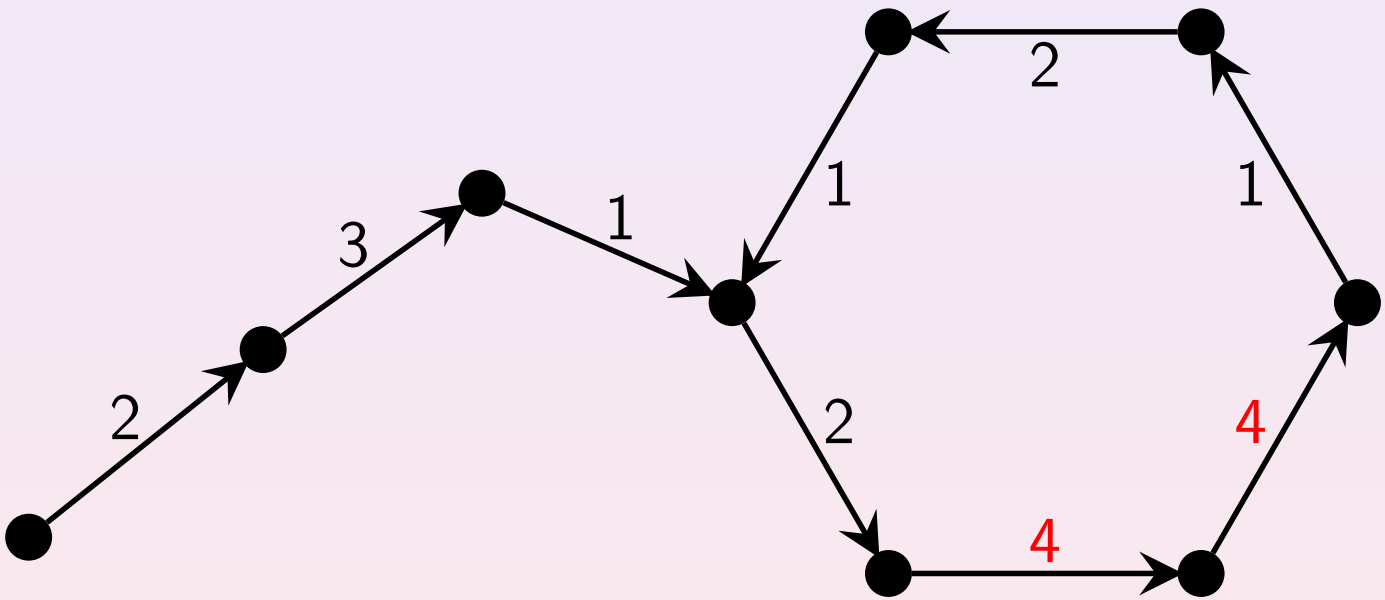
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $231(244121)^\infty$

Automate réduit (2)

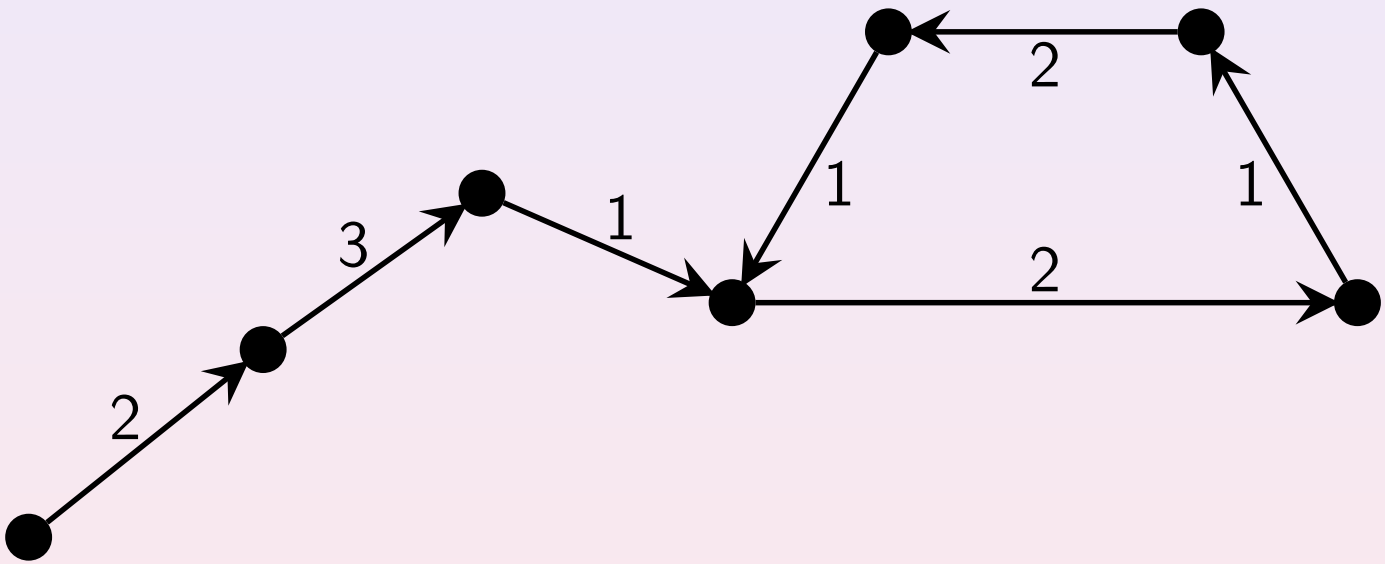
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $231(244121)^\infty$

Automate réduit (2)

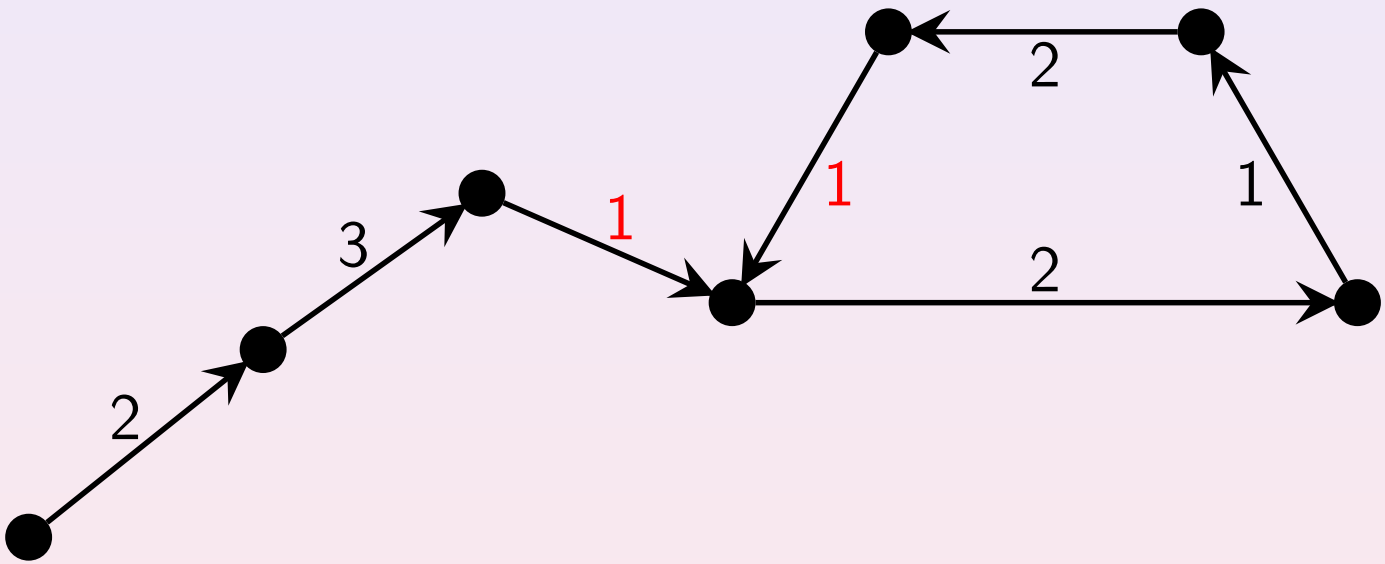
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $231(2121)^\infty$

Automate réduit (2)

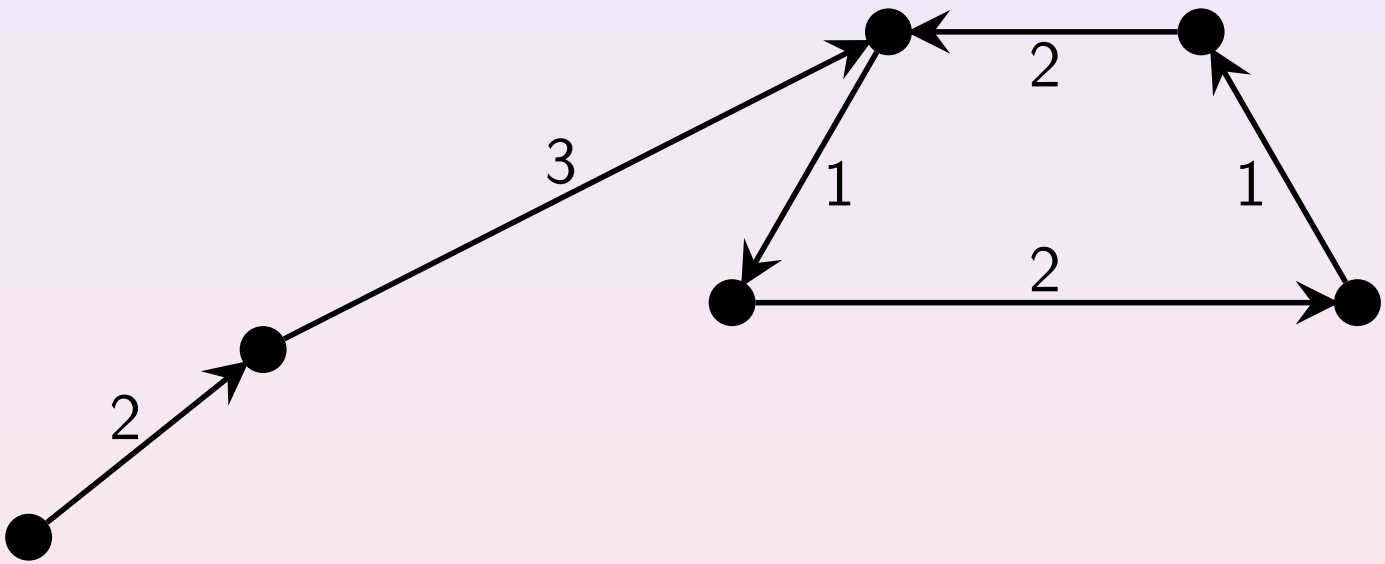
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $231(2121)^\infty$

Automate réduit (2)

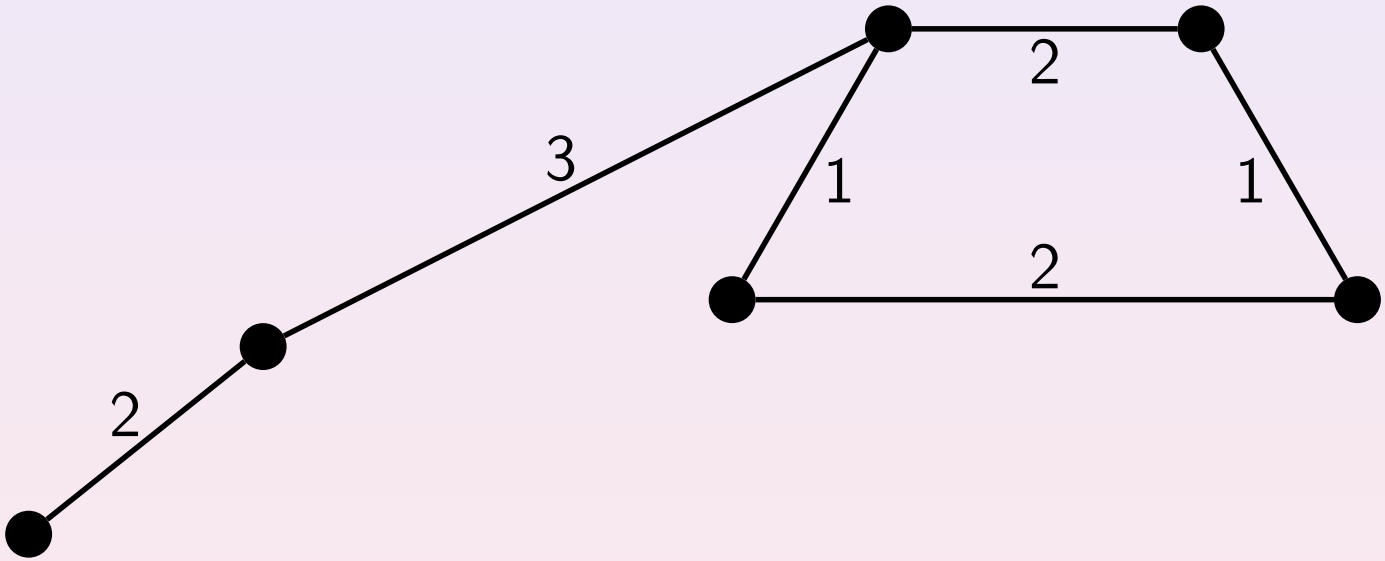
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $23(1212)^\infty$

Automate réduit (2)

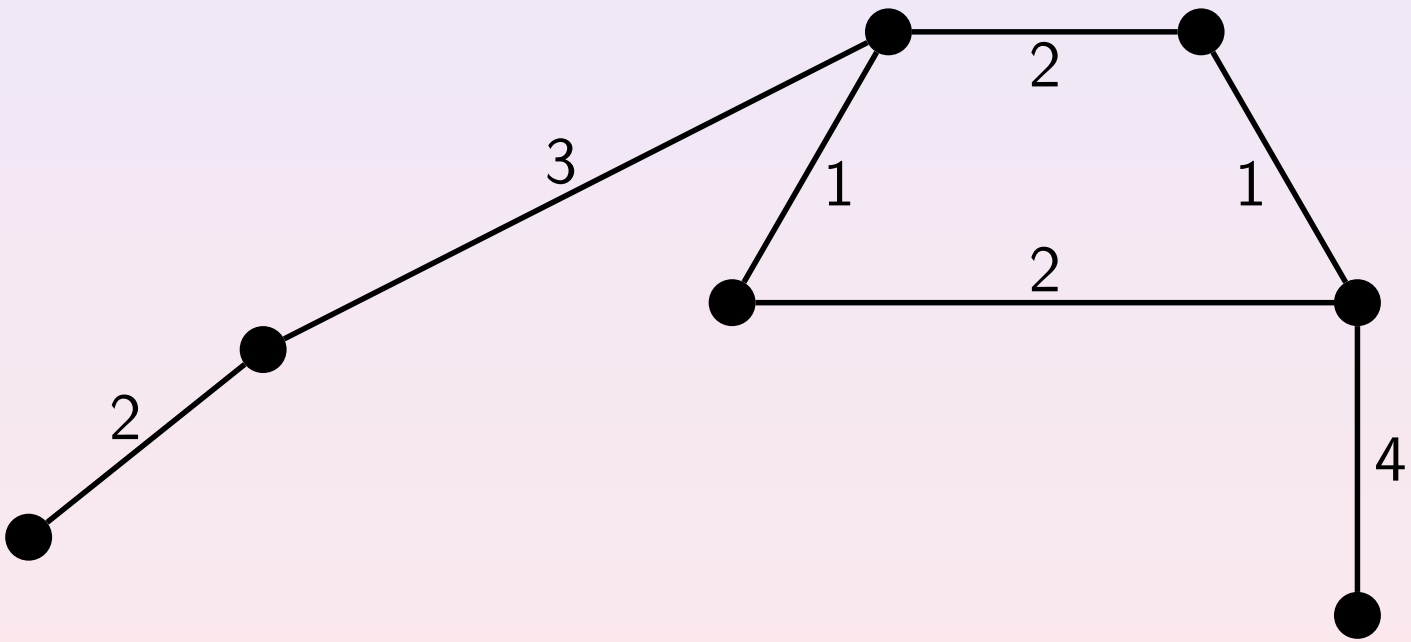
Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $23(1212)^\infty$

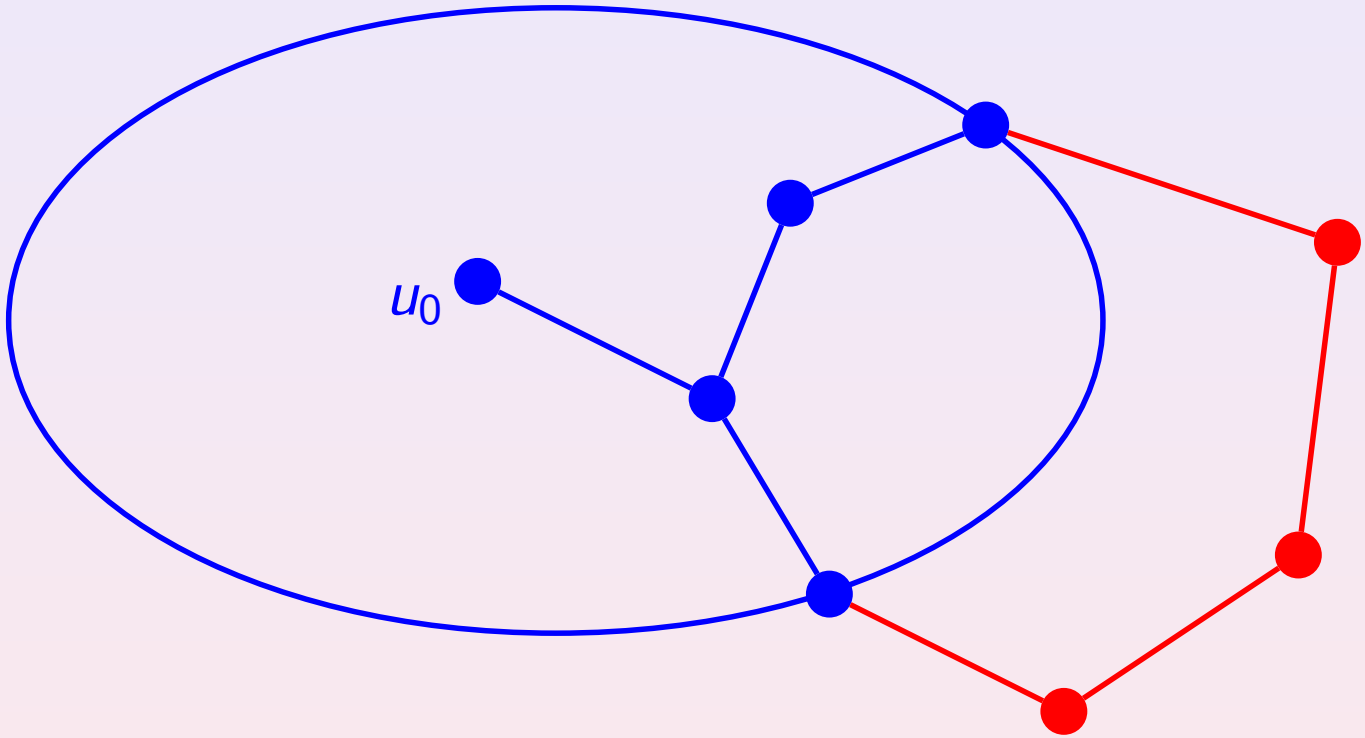
Automate réduit (2)

Dans les graphes d-homogènes, les deux déplacements i, i forment un cycle (aller-retour le long d'une arête).



Empreinte de l'automate : $231(244121)^\infty$
Empreinte de l'automate réduit : $23(1212)^\infty$

Récurrence



Piège pour q automates non coopératifs à K états : $O(qK)$ sommets.

Exploration avec arrêt (1)

Objectif (rappel)

Concevoir un piège pour un automate A à K états muni d'un caillou.

Notations

- A_0 : A sans le caillou est équivalent à un automate simple à K états A_0
- A_1 : A toujours en possession du caillou est équivalent à un automate simple à K états A_1

Exploration avec arrêt (2)

- Si le caillou est toujours proche de l'automate
  - Pas vraiment plus puissant qu'un automate simple
  - Piège pour A_1 comme méta-structure $\Longrightarrow$ $O(K)$ sommets
- Si l'automate est loin du caillou
  - Pièges pour les automates simples (comme A_0)
  - L'automate peut entrer dans les pièges dans différents états
  - Piège pour les K différents automates ayant la même fonction de transition que A_0 $\Longrightarrow$ $O(K^2)$ sommets

Taille totale du piège : $O(K^3)$ sommets

Exploration avec arrêt (2)

- Si le caillou est toujours proche de l'automate
  - Pas vraiment plus puissant qu'un automate simple
  - Piège pour A_1 comme méta-structure $\Longrightarrow$ $O(K)$ sommets
- Si l'automate est loin du caillou
  - Pièges pour les automates simples (comme A_0)
  - L'automate peut entrer dans les pièges dans différents états
  - Piège pour les K différents automates ayant la même fonction de transition que A_0 $\Longrightarrow$ $O(K^2)$ sommets

Taille totale du piège : $O(K^3)$ sommets

Exploration avec arrêt (2)

- Si le caillou est toujours proche de l'automate
  - Pas vraiment plus puissant qu'un automate simple
  - Piège pour A_1 comme méta-structure $\Longrightarrow$ $O(K)$ sommets
- Si l'automate est loin du caillou
  - Pièges pour les automates simples (comme A_0)
  - L'automate peut entrer dans les pièges dans différents états
  - Piège pour les K différents automates ayant la même fonction de transition que A_0 $\Longrightarrow$ $O(K^2)$ sommets

Taille totale du piège : $O(K^3)$ sommets

Plan

Aider l'automate

- Choisir les numéros de port
- Donner de l'information à l'automate
- Mettre de petites étiquettes sur les sommets

Résultats de la thèse (3)

Question

Quelle est la fonction minimale $\pi(n)$ telle qu'il existe un algorithme fixant les numéros de port et un automate fini les utilisant pour explorer périodiquement tous les graphes de taille n avec une période au plus $\pi(n)$?

Numéros de port

En choisissant convenablement les numéros de port, il existe un automate à trois états explorant tous les graphes en au plus $4n$ étapes.

Résultats de la thèse (3)

Question

Quelle est la fonction minimale $\pi(n)$ telle qu'il existe un algorithme fixant les numéros de port et un automate fini les utilisant pour explorer périodiquement tous les graphes de taille n avec une période au plus $\pi(n)$?

Numéros de port

En choisissant convenablement les numéros de port, il existe un automate à trois états explorant tous les graphes en au plus $4n$ étapes.

Définition du conseil

Définition

- Un oracle donne un conseil au robot sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Énoncé du problème

Quelle est la taille minimale d'un conseil permettant l'exploration sous certaines contraintes de performance ?

Définition du conseil

Définition

- Un oracle donne un conseil au robot sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Énoncé du problème

Quelle est la taille minimale d'un conseil permettant l'exploration sous certaines contraintes de performance ?

Définition du conseil

Définition

- Un oracle donne un conseil au robot sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Énoncé du problème

Quelle est la taille minimale d'un conseil permettant l'exploration sous certaines contraintes de performance ?

Exploration d'arbres par un robot

Objectif

Visiter tous les sommets le plus rapidement possible

Mesure

Rapport compétitif d'un algorithme $\mathcal{A}$:

$$\frac{\text{longueur du chemin suivi par } \mathcal{A}}{\text{plus court chemin couvrant l'arbre}}$$

(maximisé sur tous les arbres et tous les sommets de départ)

Exploration d'arbres par un robot

Objectif

Visiter tous les sommets le plus rapidement possible

Mesure

Rapport compétitif d'un algorithme $\mathcal{A}$:

$$\frac{\text{longueur du chemin suivi par } \mathcal{A}}{\text{plus court chemin couvrant l'arbre}}$$

(maximisé sur tous les arbres et tous les sommets de départ)

Résultats de la thèse (4)

Dessmark, Pelc, TCS 2004
Optimal graph exploration without good maps

DFS a un rapport compétitif égal à 2

- Aucune information
  - Aucun algorithme ne peut battre le DFS
- Connaissance du graphe (non étiqueté)
  - Rapport compétitif strictement inférieur à 2 atteignable

Résultat

Seuil sur la taille du conseil $\approx \log \log D$ bits (D = diamètre)

Résultats de la thèse (4)

Dessmark, Pelc, TCS 2004
Optimal graph exploration without good maps

DFS a un rapport compétitif égal à 2

- Aucune information
  - Aucun algorithme ne peut battre le DFS
- Connaissance du graphe (non étiqueté)
  - Rapport compétitif strictement inférieur à 2 atteignable

Résultat

Seuil sur la taille du conseil $\approx \log \log D$ bits (D = diamètre)

Résultats de la thèse (5)

Problème

Taille minimale d'un conseil permettant l'exploration par un automate fini des graphes arbitraires.

Modèle

- Un oracle colorie (étiquette) le graphe pour aider l'automate.
- L'automate peut lire la couleur comme entrée de sa fonction de transition.

Résultats

Trois couleurs sont suffisantes.

Résultats de la thèse (5)

Problème

Taille minimale d'un conseil permettant l'exploration par un automate fini des graphes arbitraires.

Modèle

- Un oracle colorie (étiquette) le graphe pour aider l'automate.
- L'automate peut lire la couleur comme entrée de sa fonction de transition.

Résultats

Trois couleurs sont suffisantes.

Résultats de la thèse (5)

Problème

Taille minimale d'un conseil permettant l'exploration par un automate fini des graphes arbitraires.

Modèle

- Un oracle colorie (étiquette) le graphe pour aider l'automate.
- L'automate peut lire la couleur comme entrée de sa fonction de transition.

Résultats

Trois couleurs sont suffisantes.

Résultats (Cohen, Fraigniaud, I., Korman, Peleg)

Théorème

Il existe un automate fini et un algorithme de coloriage en seulement trois couleurs tels que l'automate peut explorer tous les graphes coloriés par l'algorithme.

Automate de Mealy

Entrées

- S : état courant
- i : port d'entrée
- d : degré du nœud

Fonction de transition/sortie

$$(S', j) = \mu(S, i, d)$$

Sorties

- S' : nouvel état
- j : port de sortie

Automate de Mealy

Entrées

- S : état courant
- i : port d'entrée
- d : degré du nœud

Fonction de transition/sortie

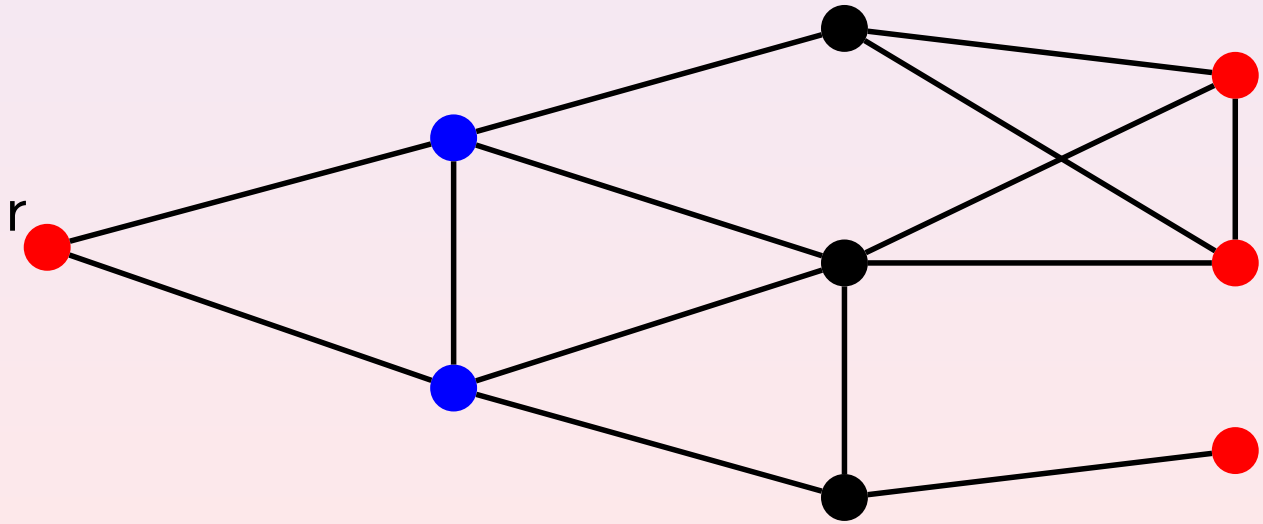
$(S', j) = \mu\ (S, i, d)$

Sorties

- S' : nouvel état
- j : port de sortie

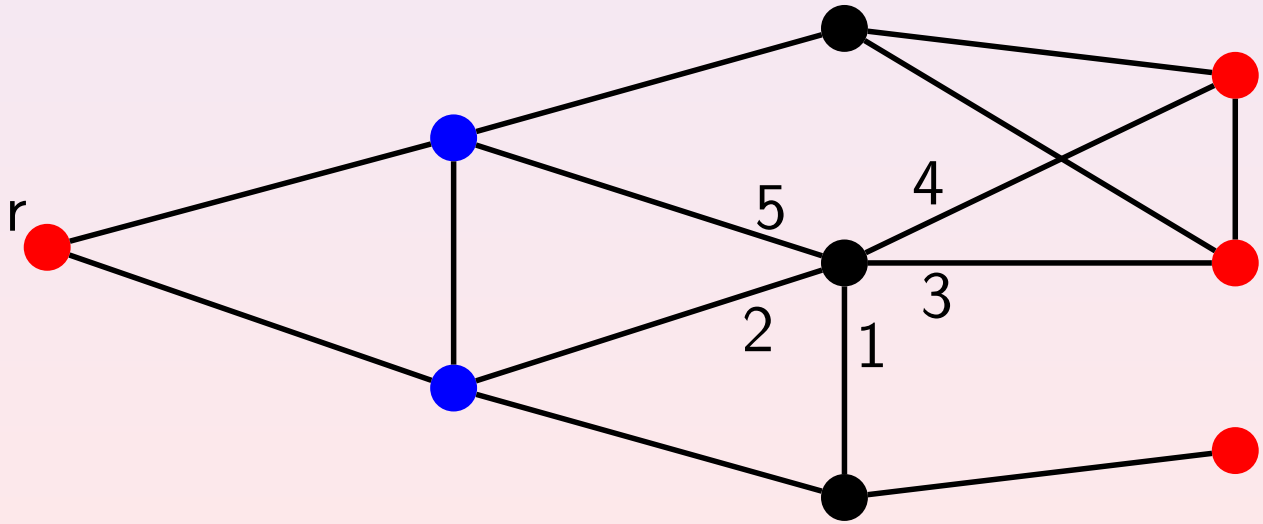
Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
  - distance $d \equiv 0[3]$ rouge
  - distance $d \equiv 1[3]$ bleu
  - distance $d \equiv 2[3]$ noir



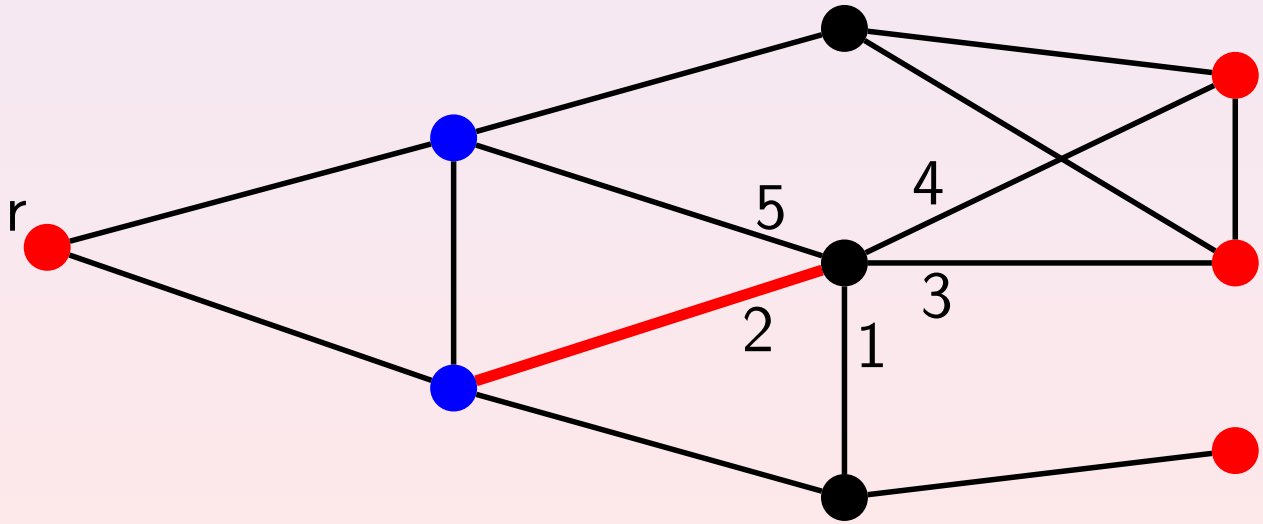
Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
  - distance $d \equiv 0[3]$ rouge
  - distance $d \equiv 1[3]$ bleu
  - distance $d \equiv 2[3]$ noir



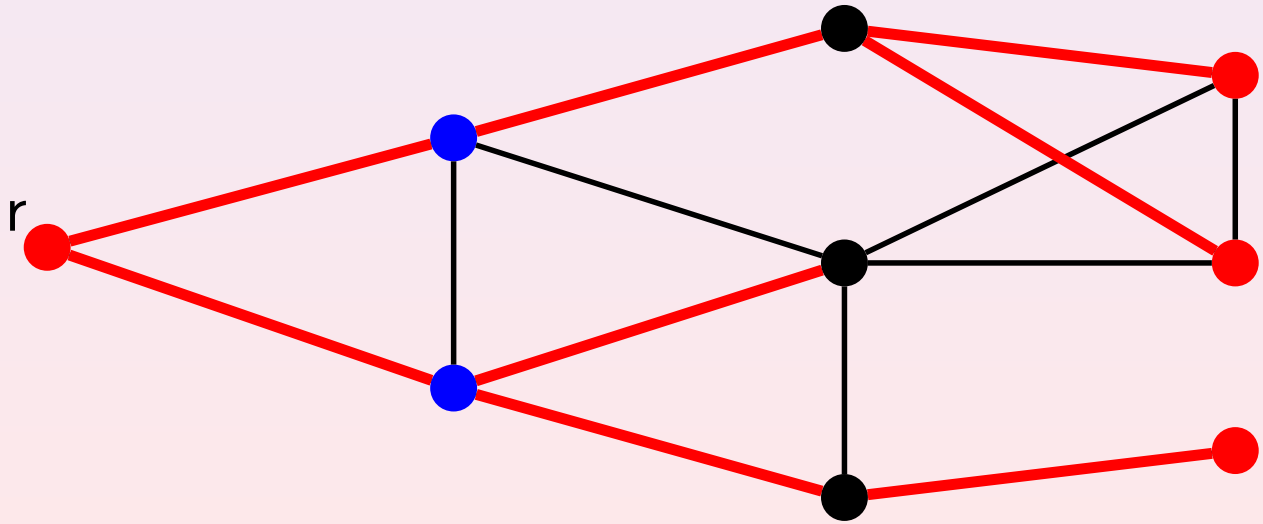
Trois couleurs suffisent

- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
  - distance $d \equiv 0[3]$ rouge
  - distance $d \equiv 1[3]$ bleu
  - distance $d \equiv 2[3]$ noir

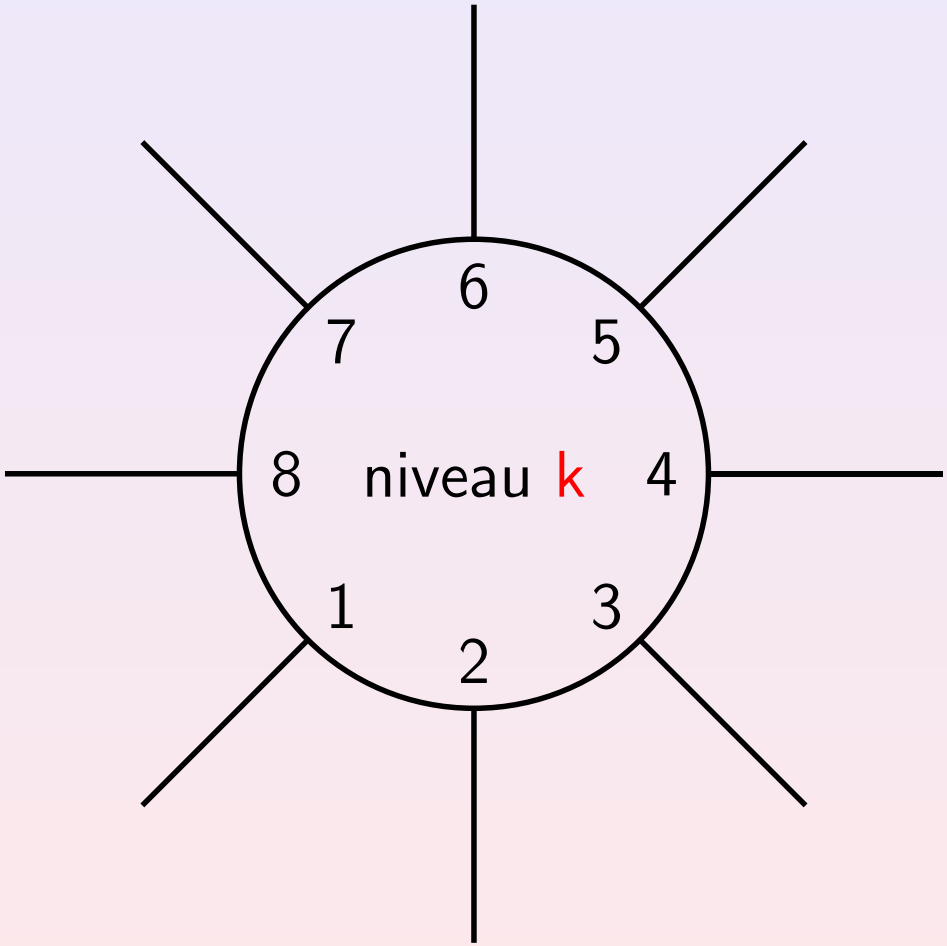


Trois couleurs suffisent

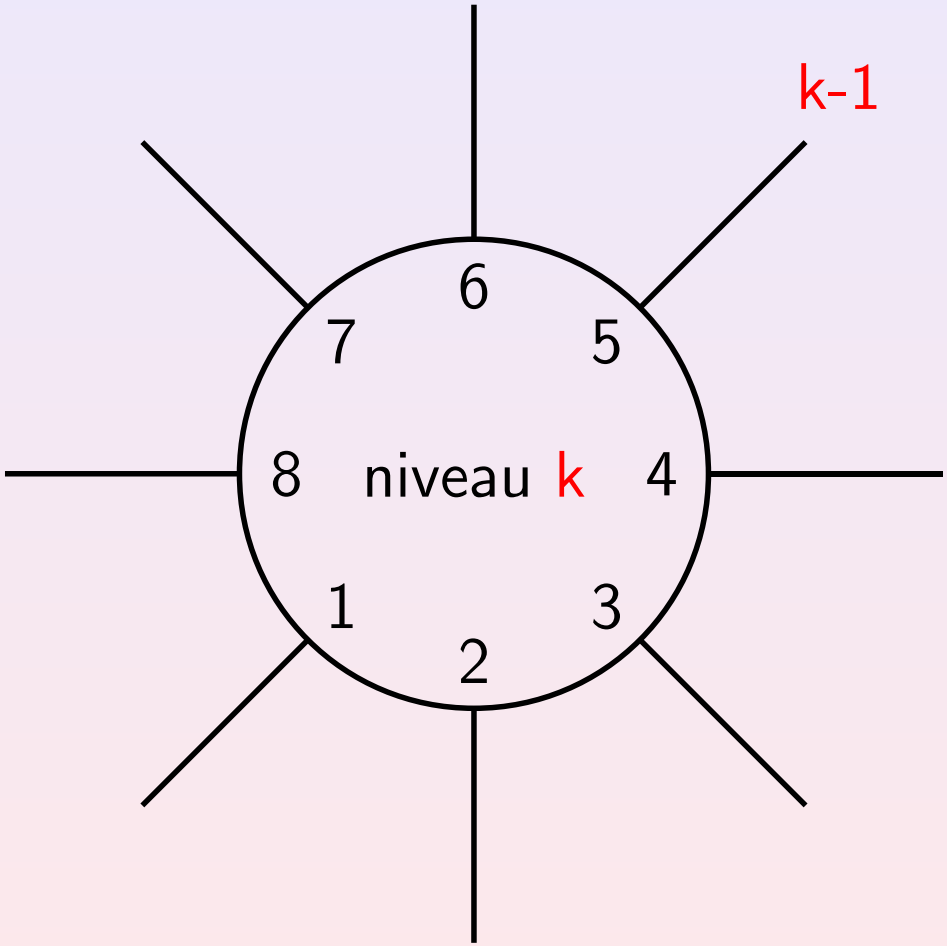
- choisir arbitrairement un nœud comme racine
- colorier tous les nœuds en fonction de leur distance d à la racine
  - distance $d \equiv 0[3]$ rouge
  - distance $d \equiv 1[3]$ bleu
  - distance $d \equiv 2[3]$ noir



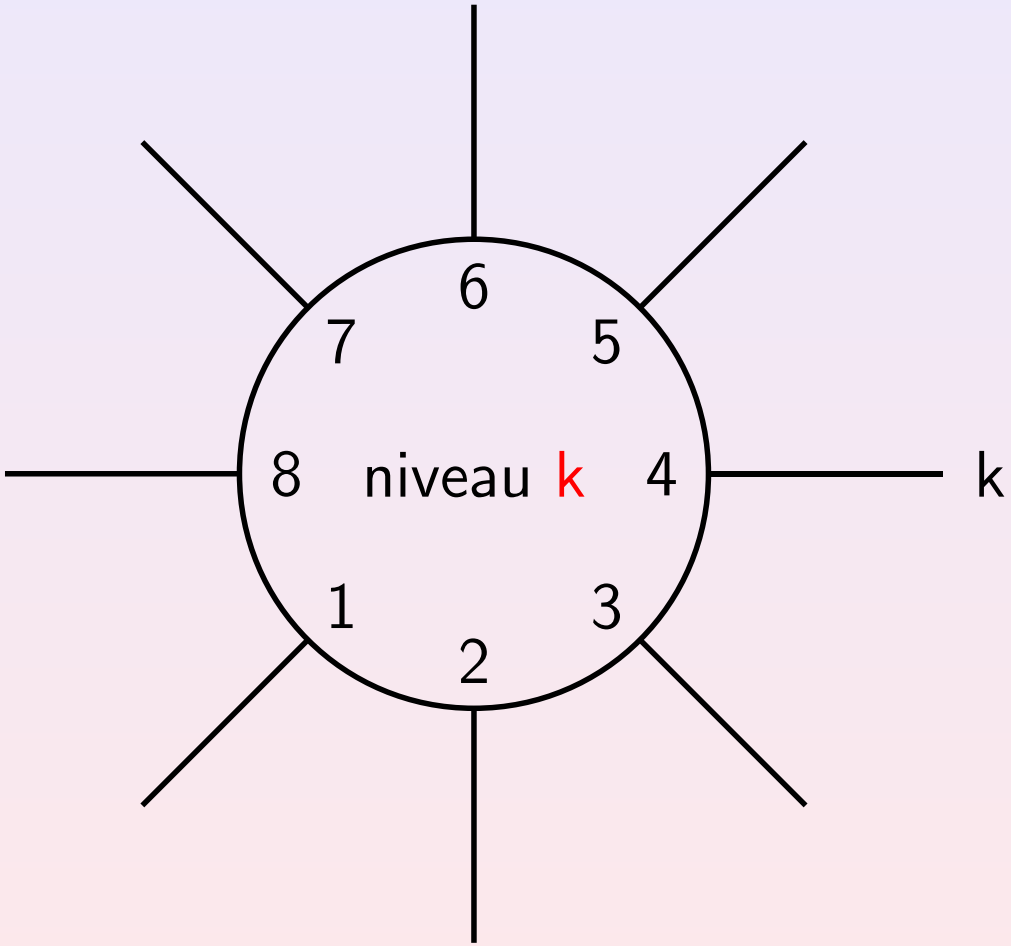
Reconnaissance du père



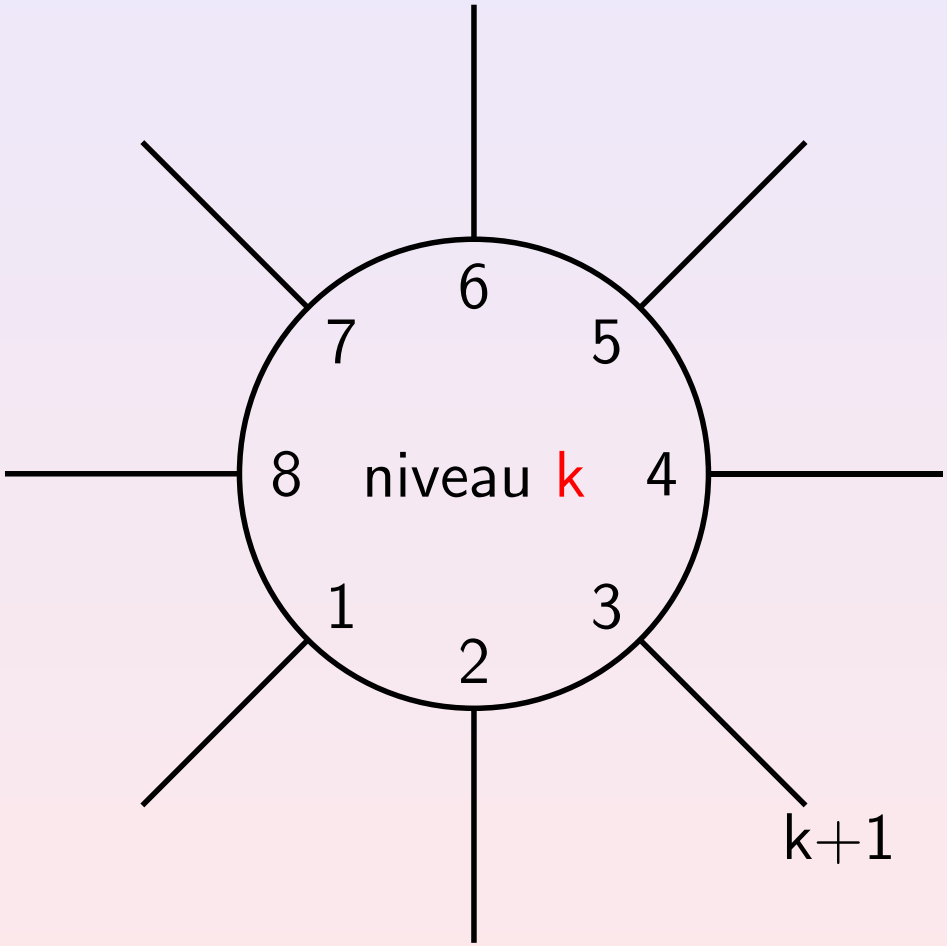
Reconnaissance du père



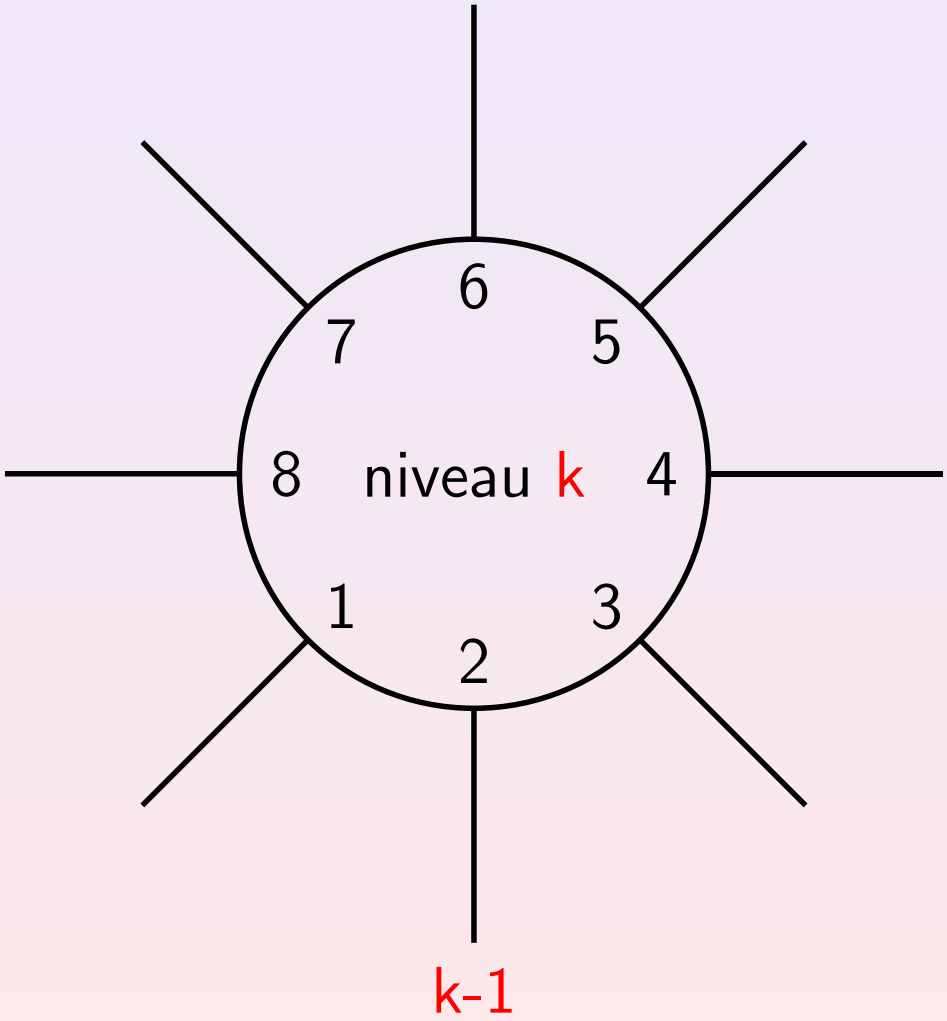
Reconnaissance du père



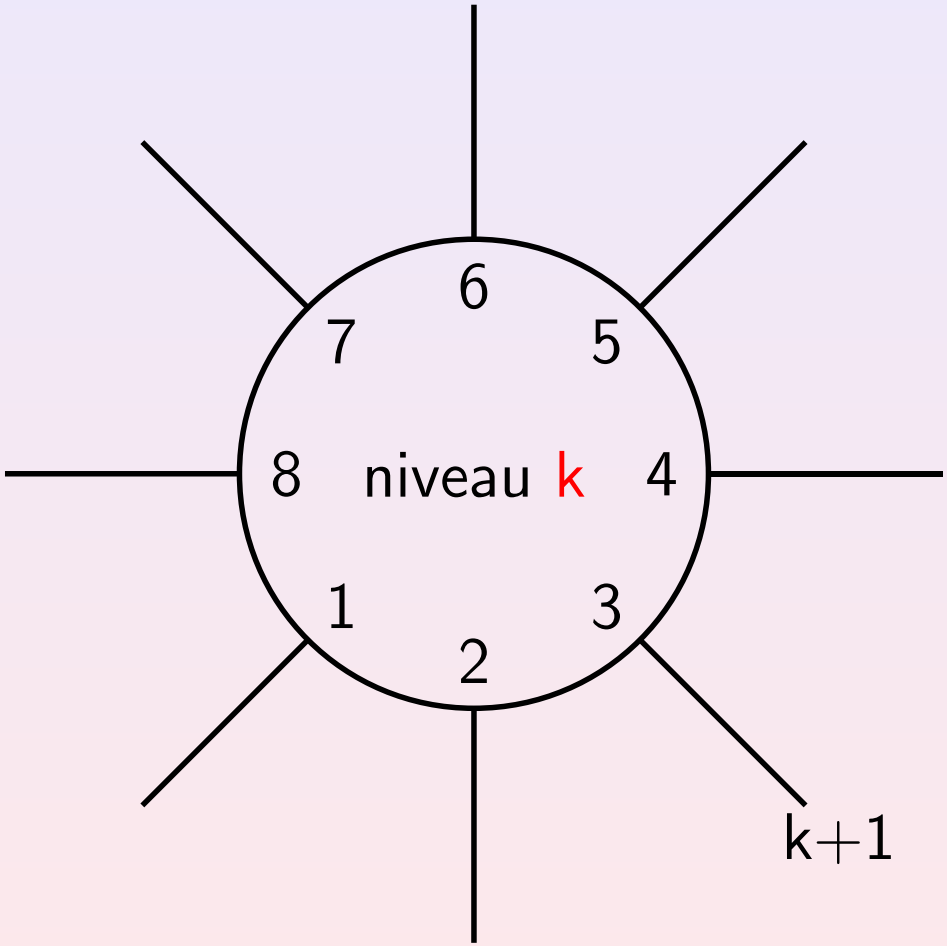
Reconnaissance du père



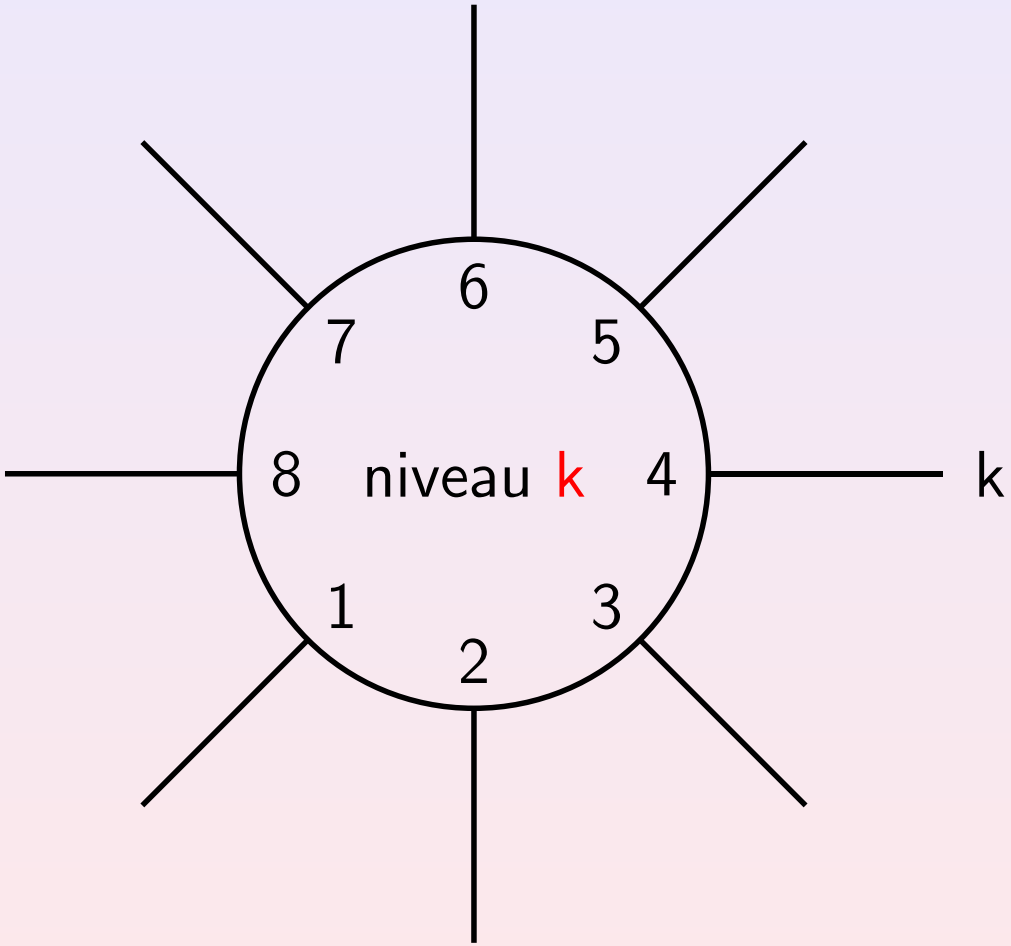
Reconnaissance du père



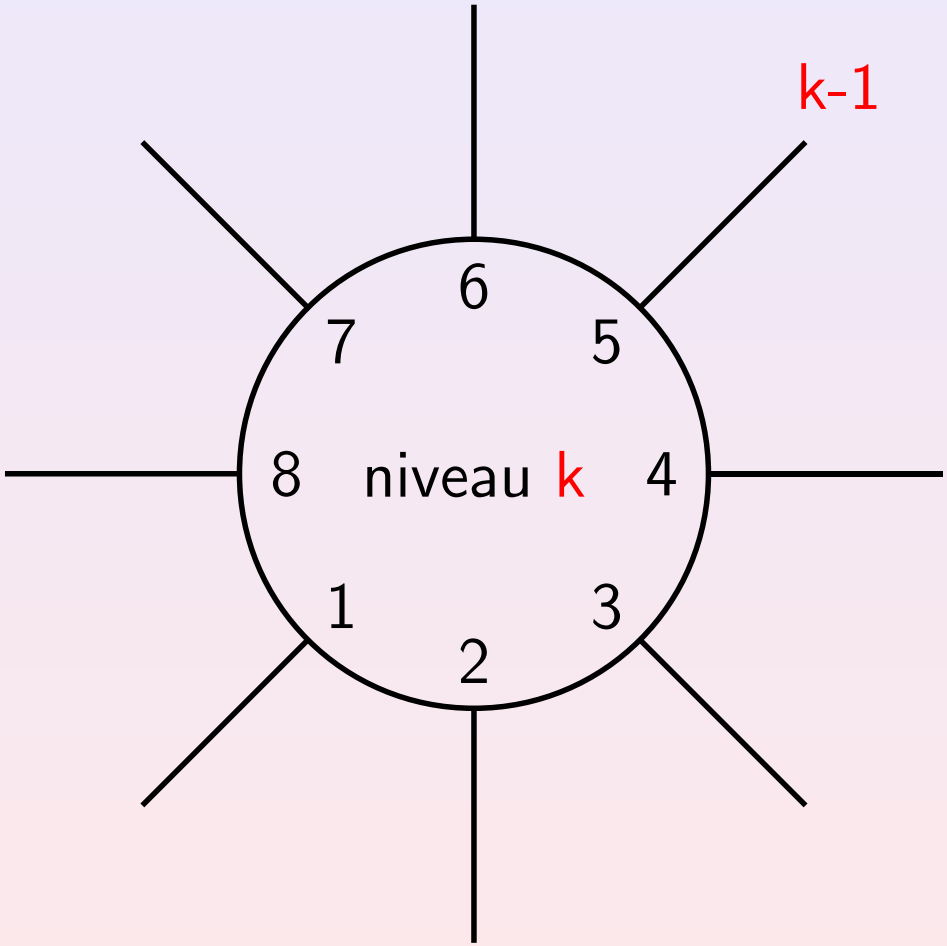
Reconnaissance du père



Reconnaissance du père



Reconnaissance du père



Plan

Résumé des résultats obtenus

Exploration sans assistance

- Bornes sur la mémoire (orientés et non orientés)
- Hiérarchie des automates

Exploration avec assistance

- Exploration par automate fini
  - choix des numéros de port
  - coloriage des sommets
- Introduction de la notion de conseil

Résumé des résultats obtenus

Exploration sans assistance

- Bornes sur la mémoire (orientés et non orientés)
- Hiérarchie des automates

Exploration avec assistance

- Exploration par automate fini
  - choix des numéros de port
  - coloriage des sommets
- Introduction de la notion de conseil

Conclusion

Exploration avec arrêt des graphes orientés

- Borne inf. : $\Omega(n \log \Delta)$ bits
- Borne sup. : $O(n\Delta \log n)$ bits (taille d'une carte)

Problème ouvert :

- L'exploration nécessite-t-elle strictement moins de mémoire que la cartographie ?

Coloriage des sommets par un oracle

- Deux couleurs sont nécessaires.
- Trois couleurs suffisent.

Problème ouvert :

- Deux couleurs suffisent-elles pour les graphes arbitraires ?

Conclusion

Exploration avec arrêt des graphes orientés

- Borne inf. : $\Omega(n \log \Delta)$ bits
- Borne sup. : $O(n\Delta \log n)$ bits (taille d'une carte)

Problème ouvert :

- L'exploration nécessite-t-elle strictement moins de mémoire que la cartographie ?

Coloriage des sommets par un oracle

- Deux couleurs sont nécessaires.
- Trois couleurs suffisent.

Problème ouvert :

- Deux couleurs suffisent-elles pour les graphes arbitraires ?

Conclusion

Exploration avec arrêt des graphes orientés

- Borne inf. : $\Omega(n \log \Delta)$ bits
- Borne sup. : $O(n\Delta \log n)$ bits (taille d'une carte)

Problème ouvert :

- L'exploration nécessite-t-elle strictement moins de mémoire que la cartographie ?

Coloriage des sommets par un oracle

- Deux couleurs sont nécessaires.
- Trois couleurs suffisent.

Problème ouvert :

- Deux couleurs suffisent-elles pour les graphes arbitraires ?

Conclusion

Exploration avec arrêt des graphes orientés

- Borne inf. : $\Omega(n \log \Delta)$ bits
- Borne sup. : $O(n\Delta \log n)$ bits (taille d'une carte)

Problème ouvert :

- L'exploration nécessite-t-elle strictement moins de mémoire que la cartographie ?

Coloriage des sommets par un oracle

- Deux couleurs sont nécessaires.
- Trois couleurs suffisent.

Problème ouvert :

- Deux couleurs suffisent-elles pour les graphes arbitraires ?

Perspectives

Observation

L'efficacité des solutions à un problème dépend souvent de la connaissance donnée a priori sur la topologie.

Perspectives

Étude des compromis performance/connaissance en calcul distribué grâce au concept de conseil donné par un oracle.

Perspectives

Observation

L'efficacité des solutions à un problème dépend souvent de la connaissance donnée a priori sur la topologie.

Perspectives

Étude des compromis performance/connaissance en calcul distribué grâce au concept de conseil donné par un oracle.

Quelques exemples

Exploration des graphes orientés en temps polynomial [BF+02]

- Impossible si aucune information
- Possible si connaissance d'une borne sup. sur n

Diffusion déterministe synchrone dans les réseaux radio

- Temps $\Omega(n \log D)$ si aucune information [CMS01]
- Temps $O(D + \log^2 n)$ si connaissance complète [KP06]

Réveil (wakeup) dans les réseaux filaires [AGPV90]

Connaissance du voisinage à distance ρ

- $O(n^{1+\Theta(1)/\rho})$ messages de taille bornée

Quelques exemples

Exploration des graphes orientés en temps polynomial [BF+02]

- Impossible si aucune information
- Possible si connaissance d'une borne sup. sur n

Diffusion déterministe synchrone dans les réseaux radio

- Temps $\Omega(n \log D)$ si aucune information [CMS01]
- Temps $O(D + \log^2 n)$ si connaissance complète [KP06]

Réveil (wakeup) dans les réseaux filaires [AGPV90]

Connaissance du voisinage à distance ρ

- $O(n^{1+\Theta(1)/\rho})$ messages de taille bornée

Quelques exemples

Exploration des graphes orientés en temps polynomial [BF+02]

- Impossible si aucune information
- Possible si connaissance d'une borne sup. sur n

Diffusion déterministe synchrone dans les réseaux radio

- Temps $\Omega(n \log D)$ si aucune information [CMS01]
- Temps $O(D + \log^2 n)$ si connaissance complète [KP06]

Réveil (wakeup) dans les réseaux filaires [AGPV90]

Connaissance du voisinage à distance ρ

- $O(n^{1+\Theta(1)/\rho})$ messages de taille bornée

Conseil : une approche quantitative

Cadre

- Calcul distribué
- Agents mobiles

Définition

- Oracle fournit un conseil aux nœuds/agents mobiles sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Question intéressante

Quelle est la taille minimale d'un conseil permettant de résoudre le problème $\mathcal{P}$ (en un temps donné) ?

Conseil : une approche quantitative

Cadre

- Calcul distribué
- Agents mobiles

Définition

- Oracle fournit un conseil aux nœuds/agents mobiles sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Question intéressante

Quelle est la taille minimale d'un conseil permettant de résoudre le problème $\mathcal{P}$ (en un temps donné) ?

Conseil : une approche quantitative

Cadre

- Calcul distribué
- Agents mobiles

Définition

- Oracle fournit un conseil aux nœuds/agents mobiles sous la forme d'une chaîne binaire $\mathcal{O}(G)$
- Taille du conseil : $|\mathcal{O}(G)|$

Question intéressante

Quelle est la taille minimale d'un conseil permettant de résoudre le problème $\mathcal{P}$ (en un temps donné) ?

Applications (1)

Fraigniaud, I., Pelc, PODC 2006

Application du conseil à la dissémination d'informations
Contrainte : #messages $O(n)$

- réveil (wakeup) : $\Omega(n \log n)$ bits
- diffusion (broadcast) : $O(n)$ bits

$\Rightarrow$ différence clairement établie entre les deux problèmes

Applications (2)

Application du conseil au coloriage distribué

Linial, SIAM Journal on Computing, 1992

Temps minimum du 3-coloriage (sans conseil)

- $\Theta(\log^* n)$ dans les cycles
- $\Theta(D)$ dans les arbres de diamètre D

Fraigniaud, Gavoille, I., Pelc

- $\Omega(n/\log^{(cte)} n)$ bits de conseil pour temps $o(\log^* n)$ dans les cycles et $O(\log^* n)$ dans les arbres
- $\Omega(n)$ bits de conseil pour temps constant dans les cycles et dans les arbres

Applications (2)

Application du conseil au coloriage distribué

Linial, SIAM Journal on Computing, 1992

Temps minimum du 3-coloriage (sans conseil)

- $\Theta(\log^* n)$ dans les cycles
- $\Theta(D)$ dans les arbres de diamètre D

Fraigniaud, Gavoille, I., Pelc

- $\Omega(n/\log^{(cte)} n)$ bits de conseil pour temps $o(\log^* n)$ dans les cycles et $O(\log^* n)$ dans les arbres
- $\Omega(n)$ bits de conseil pour temps constant dans les cycles et dans les arbres

Merci !