Montpellier – 05/11/2009 – JGA'09

Codage des voisinages et parcours en largeur des graphes d'intervalles et de permutation

Christophe Crespelle, Philippe Gambette

Plan

- **Problème et définitions**
- **Encodage des voisinages dans les graphes d'intervalles et les graphes de permutation**
- **Recherche en largeur en $O(n)$ dans les graphes d'intervalles et les graphes de permutation**

Plan

- **Problème et définitions**
- **Encodage des voisinages dans les graphes d'intervalles et les graphes de permutation**
- **Recherche en largeur en $O(n)$ dans les graphes d'intervalles et les graphes de permutation**

Contexte

Algorithmes sur les grands graphes

- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

Contexte

Algorithmes sur les grands graphes

- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

Bon encodage vis à vis des requêtes d'adjacence :
Nombreuses classes de graphes d'intersection

Contexte

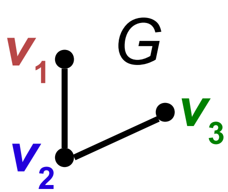
Algorithmes sur les grands graphes

- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

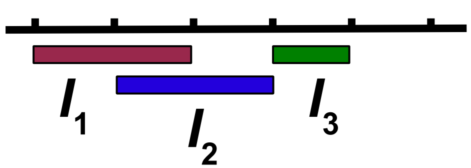
Bon encodage vis à vis des requêtes d'adjacence :

Nombreuses classes de graphes d'intersection

Exemple : graphes d'intervalles :



$\mathcal{I}$ **= {[0,2],[1,3],[3,4]}**



1 ∈ [0,2] :

1 et 2 adjacents

Encodage en O(n)

Requêtes d'adjacence en O(1)

2 sommets adjacents ssi une borne de l'intervalle de l'un est incluse dans l'intervalle de l'autre.

Contexte

Algorithmes sur les grands graphes

- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

Bon encodage vis à vis des requêtes d'adjacence :
Nombreuses classes de graphes d'intersection

Bon encodage vis à vis des requêtes de voisinage ?

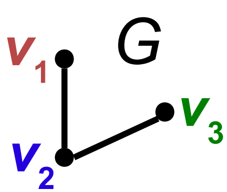
Contexte

Algorithmes sur les grands graphes

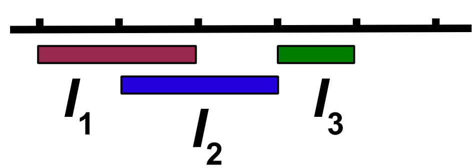
- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

Bon encodage vis à vis des requêtes d'adjacence :
Nombreuses classes de graphes d'intersection

Bon encodage vis à vis des requêtes de voisinage ?
Exemple : graphes d'intervalles :



$\mathcal{I} = \{[0,2],[1,3],[3,4]\}$



Encodage en O(n)

Requêtes de voisinage en O(n)

O(d) ?

Contexte

Algorithmes sur les grands graphes

- Stockage du graphe en mémoire — Encodage compact
- Requêtes sur le graphe — Temps de réponse réduit

Bon encodage vis à vis des requêtes d'adjacence :

Nombreuses classes de graphes d'intersection

Bon encodage vis à vis des requêtes de voisinage ?

Encodage en **O(*n*)** avec requêtes de voisinage en **O(*d*)**
pour graphes d'intervalles et graphes de permutation.
→ extension aux graphes quelconques par complétion
→ exploitation algorithmique de la structure des voisinages

Plan

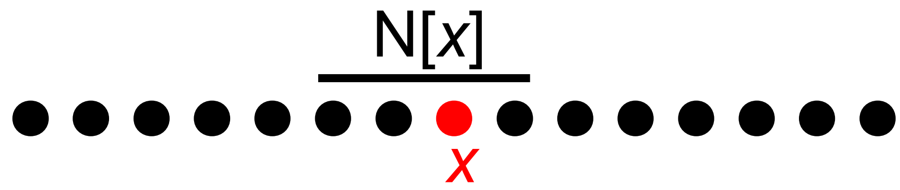
- Problème et définitions

- **Encodage des voisinages dans les graphes d'intervalles et les graphes de permutation**

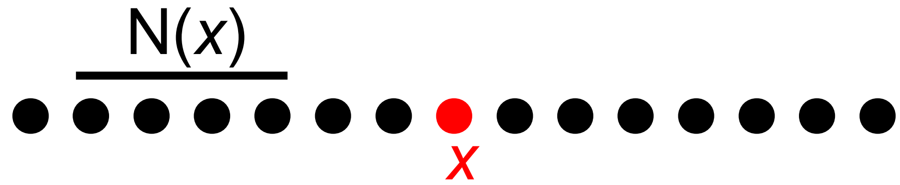
- Recherche en largeur en $O(n)$ dans les graphes d'intervalles et les graphes de permutation

Une approche naturelle

- Graphes d'intervalles propres



- Graphes biconvexes



Généralisation, 2 paramètres :

- autoriser plusieurs intervalles : **contiguïté**
- autoriser plusieurs ordres : **linéarité**

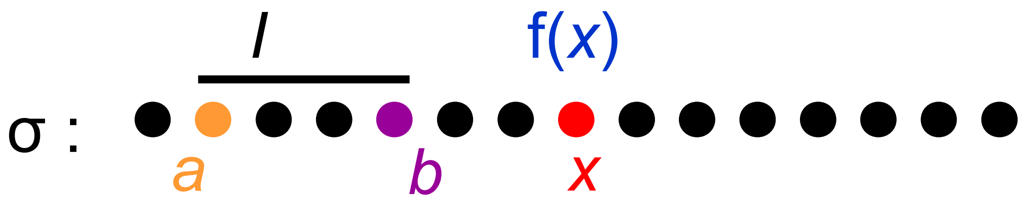
Paramètres non bornés pour graphes d'intervalles et graphes de permutation :

- contiguïté : $\Omega(\log n)$
- linéarité : $\Omega(\log n / \log \log n)$

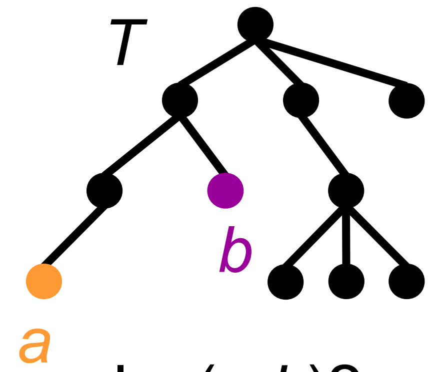
Voisinages dans les graphes de permutation

Arbres cartésiens augmentés :

Max d'une fonction entière sur un intervalle :



$$\max_{x \in I} f(x)?$$

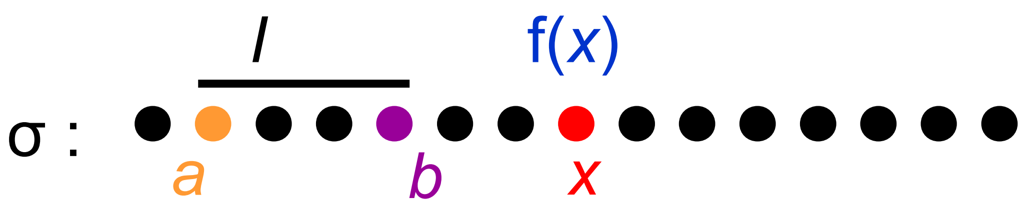


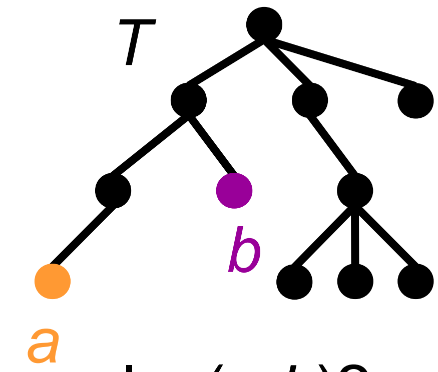
lca(a,b)?

Voisinages dans les graphes de permutation

Arbres cartésiens augmentés :

Max d'une fonction entière sur un intervalle :





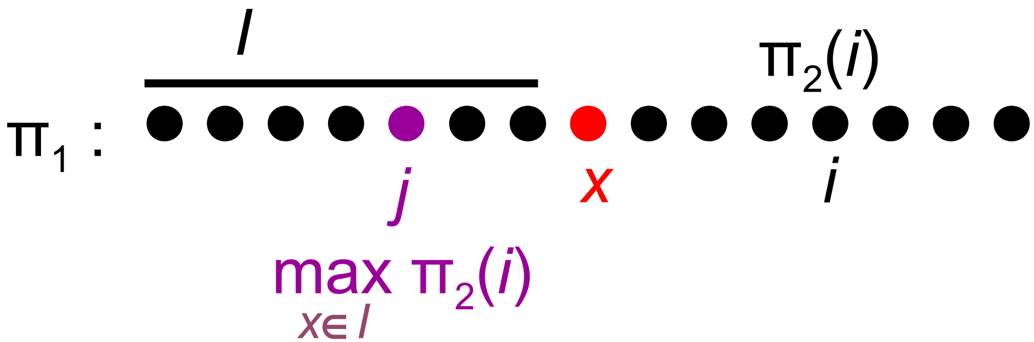
$\max_{x \in I} f(x)$?

lca(a,b)?

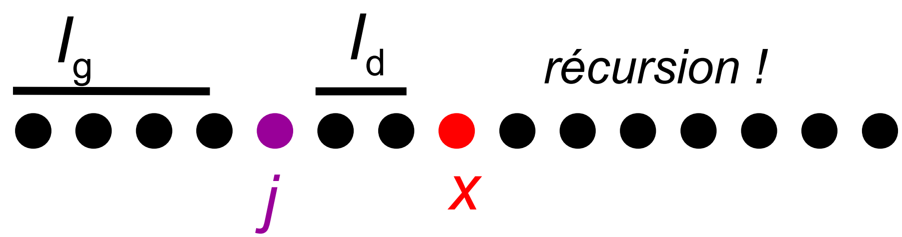
Précalcul de T en temps O(|T|)
pour répondre aux requêtes lca en O(1)

Harel & Tarjan 1984, Vuillemin 1980

Voisinages dans les graphes de permutation

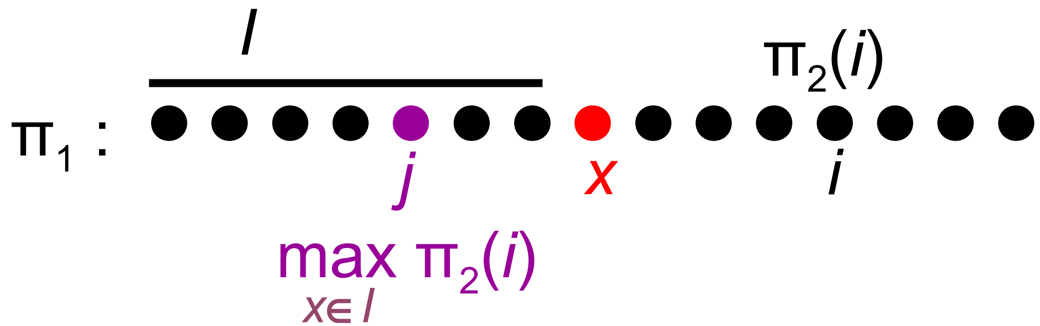


Si $\pi_2(j) > \pi_2(x)$ **alors** j est un voisin de x

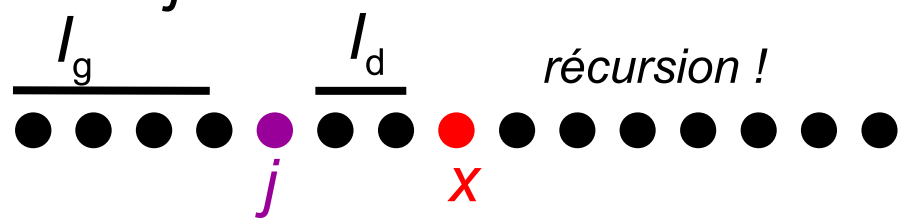


sinon x n'a pas de voisin dans I

Voisinages dans les graphes de permutation



Si $\pi_2(j) > \pi_2(x)$ **alors** j est un voisin de x



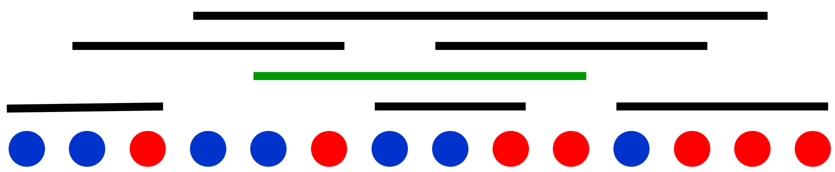
sinon x n'a pas de voisin dans I

Au plus deux appels récursif par un appel réussi
Chaque appel non réussi lancé par un appel réussi
→ complexité en temps en $O(d)$

Voisinages dans les graphes d'intervalles

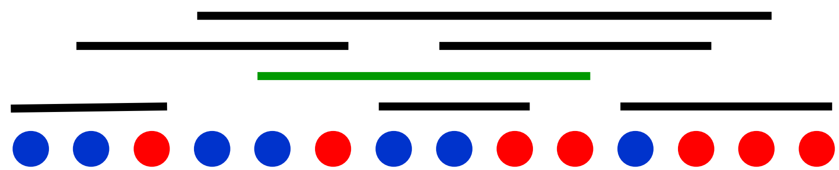
2 types de voisins :

- recouvrement
- inclusion



Voisinages dans les graphes d'intervalles

2 types de voisins :

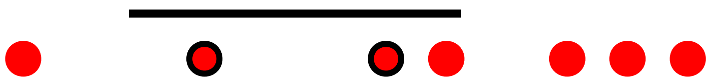


- recouvrement
- inclusion

Recouvrements :

Bornes gauches

Bornes droites



Inclusions : graphe de permutation

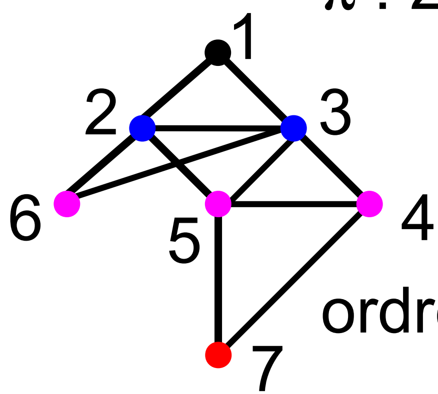
Plan

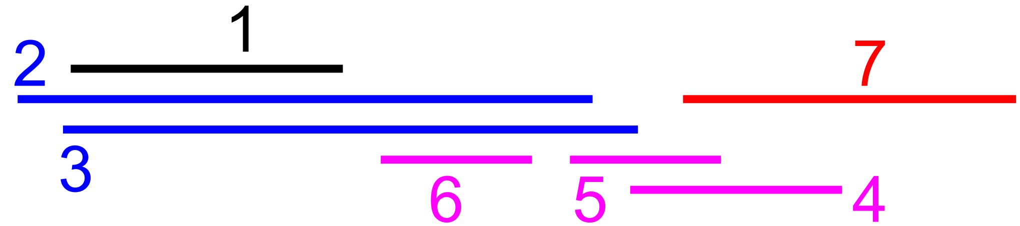
- Problème et définitions
- Encodage des voisinages dans les graphes d'intervalles et les graphes de permutation
- **Recherche en largeur en $O(n)$ dans les graphes d'intervalles et les graphes de permutation**

BFS dans les graphes d'intervalles

Entrée : ordre des bornes :

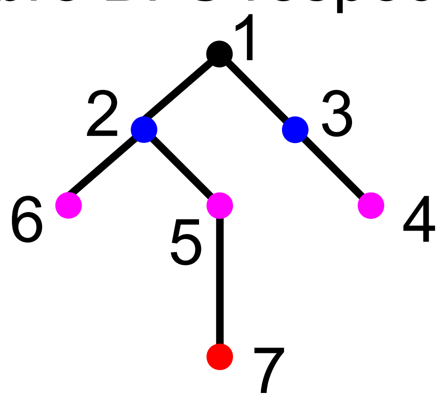
π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





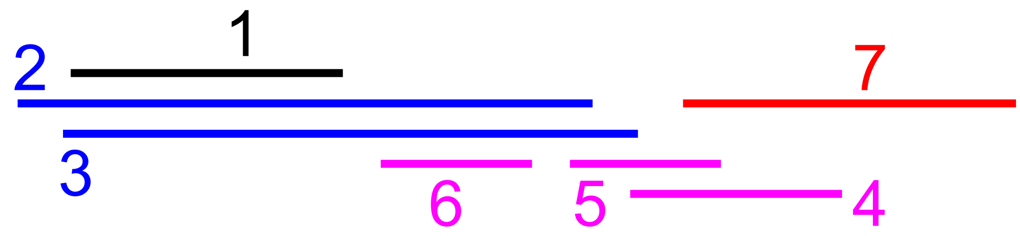
ordre de priorité σ : 1 2 3 4 5 6 7

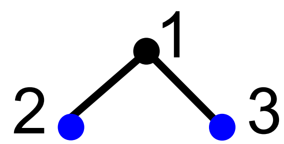
Sortie : arbre BFS respectant σ



BFS dans les graphes d'intervalles

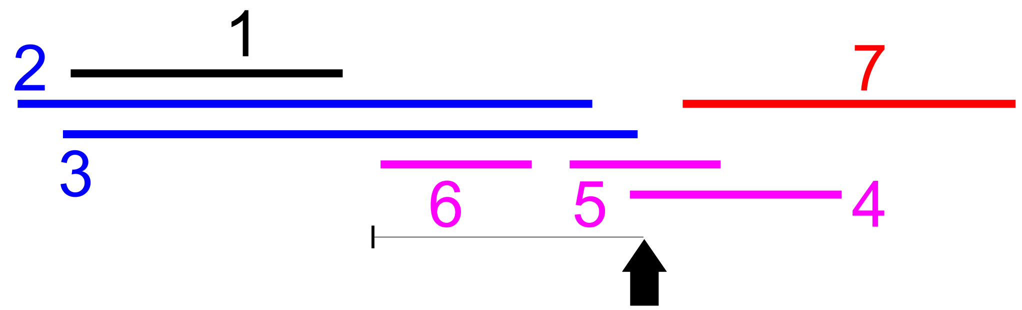
π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d

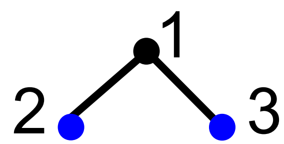




BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





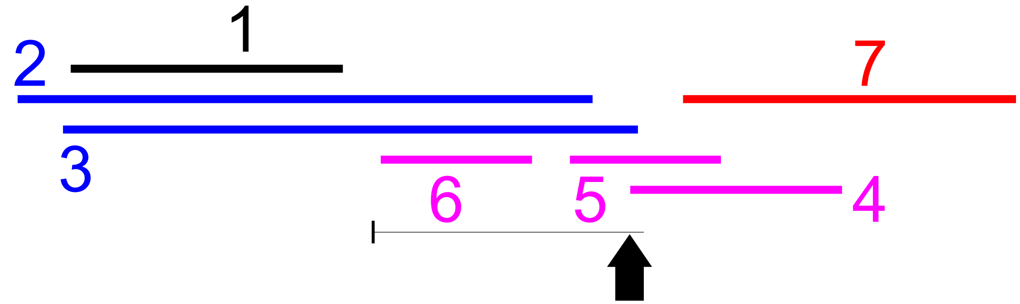
père courant : 3

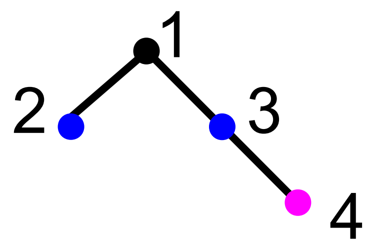
borne droite : 3

priorité' : 1

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





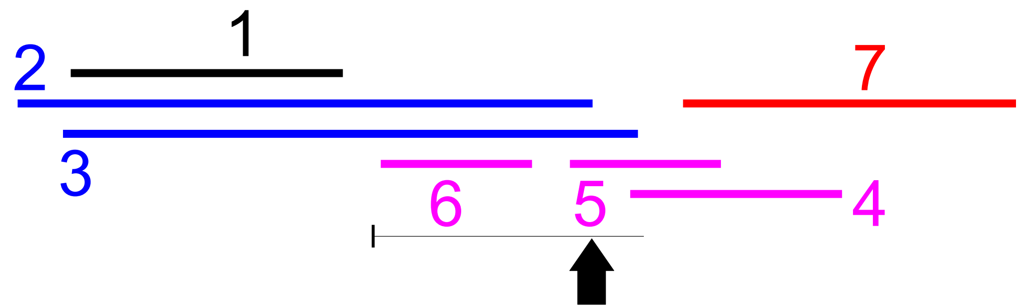
père courant : 3

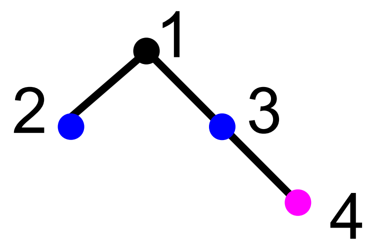
borne droite : 4

priorité' : 1 3

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





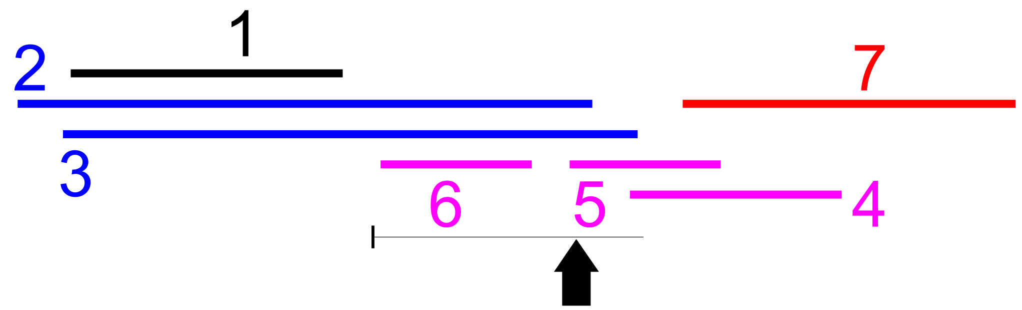
père courant : 2

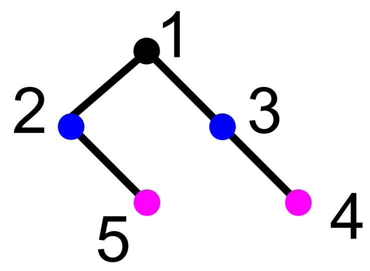
borne droite : 4

priorité' : 1 2 3

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





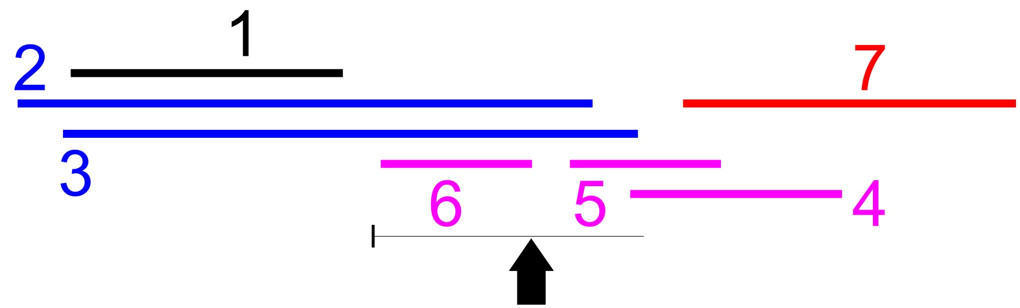
père courant : 2

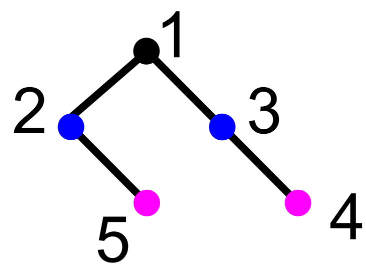
borne droite : 4

priorité' : 1 2 3

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





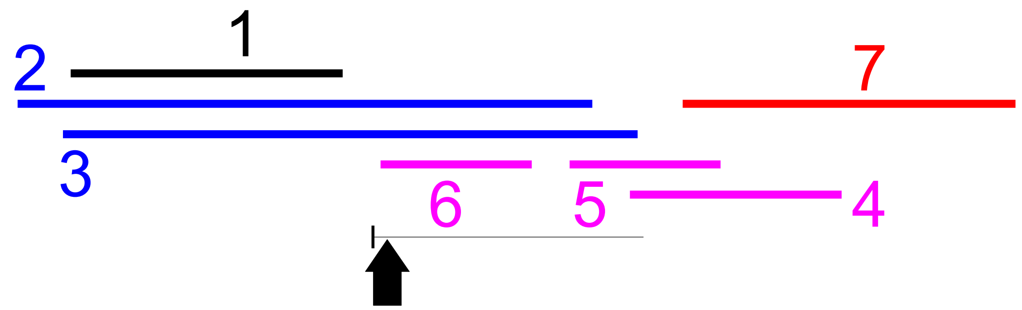
père courant : 2

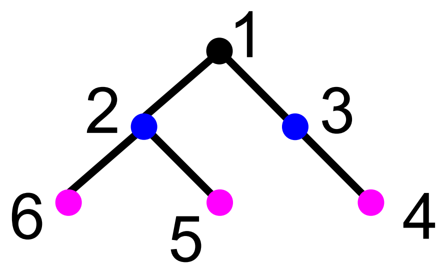
borne droite : 4

priorité' : 1 2 3

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





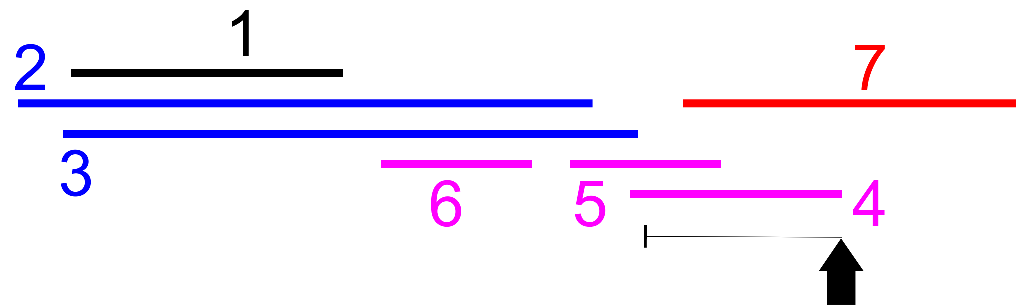
père courant : 2

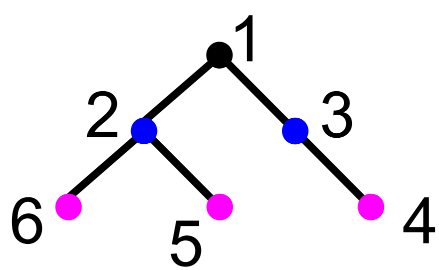
borne droite : 4

priorité' : 1 2 3

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





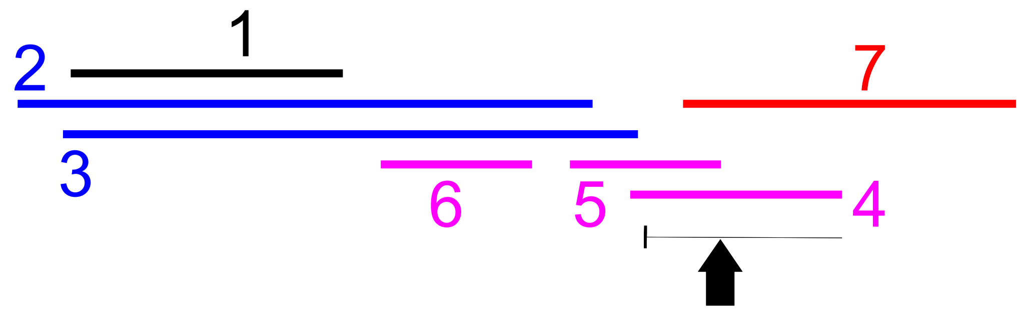
père courant : 4

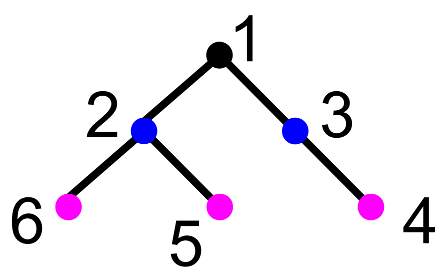
borne droite : 4

priorité' : 1 2 3 4

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





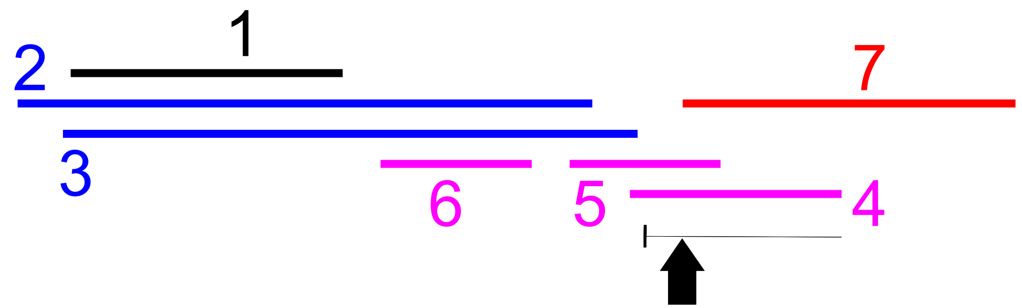
père courant : 5

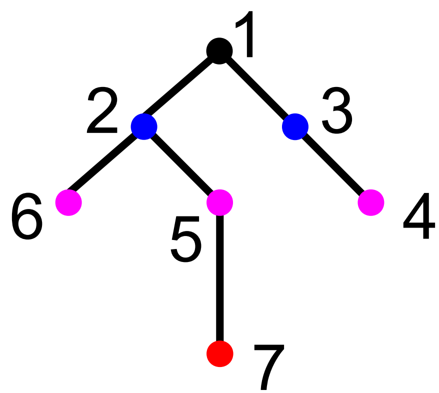
borne droite : 4

priorité' : 1 2 3 5 4

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





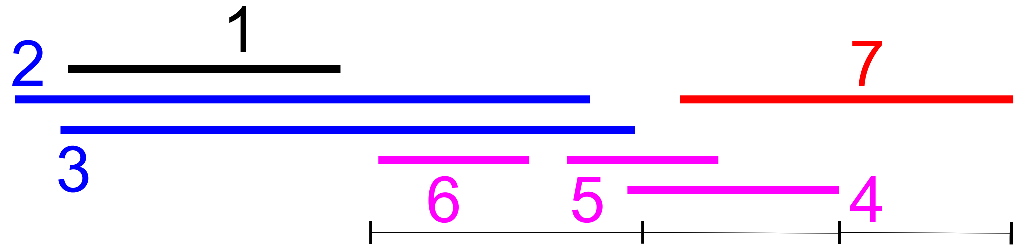
père courant : 5

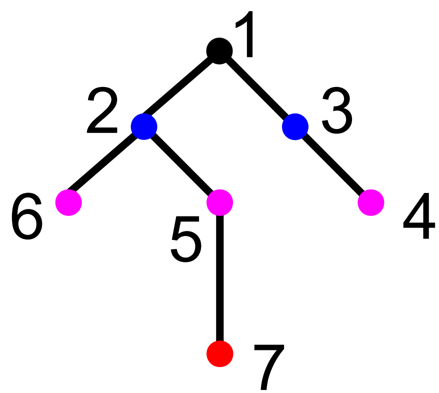
borne droite : 7

priorité' : 1 2 3 5 4

BFS dans les graphes d'intervalles

π : 2g 3g 1g 1d 6g 6d 5g 2d 4g 3d 7g 5d 4d 7d





chaque borne visitée
une seule fois :
algorithme en O(*n*)

BFS dans les graphes de permutation

Extension de l'algorithme :

- Exploration de π_1 et π_2 vers la gauche et vers la droite
- Gestion plus complexe de la liste des parents visités

Des questions ?

Merci pour votre attention !

Plus de détails dans :

• Christophe Crespelle et Philippe Gambette. Efficient Neighbourhood Encoding for Interval Graphs and Permutation Graphs and O(*n*) Breadth-First Search, *Proceedings of the 20th International Workshop on Combinatorial Algorithms* (IWOCA'09), LNCS 5874, p. 146-157.

• Christophe Crespelle et Philippe Gambette. O(*n*) Breadth First Search of Some Intersection Graph Classes. En préparation.