Partition

à la recherche de l'union

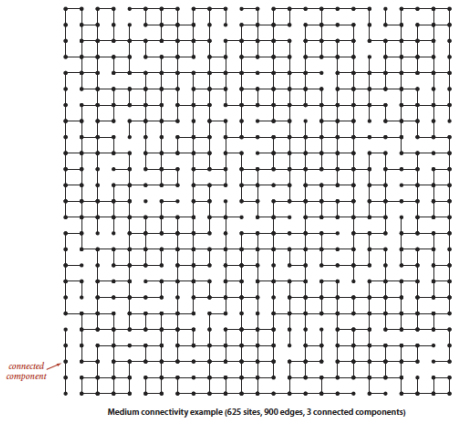
Jean-Marc.Vincent@univ-grenoble-alpes.fr[1]

[1] Laboratoire LIG
Équipe-Projet INRIA POLARIS
Université Grenoble-Alpes

Algorithmique
DIU EIL, Grenoble 2020

LE CONTEXTE

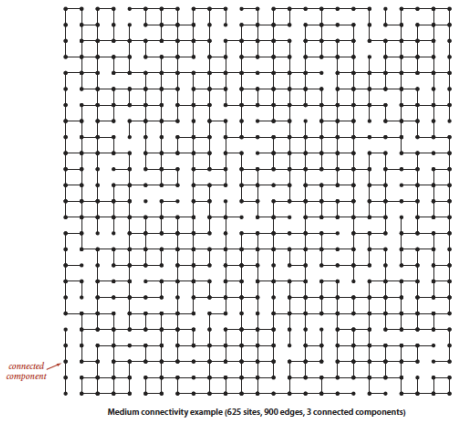
Réseau, problème de connexion

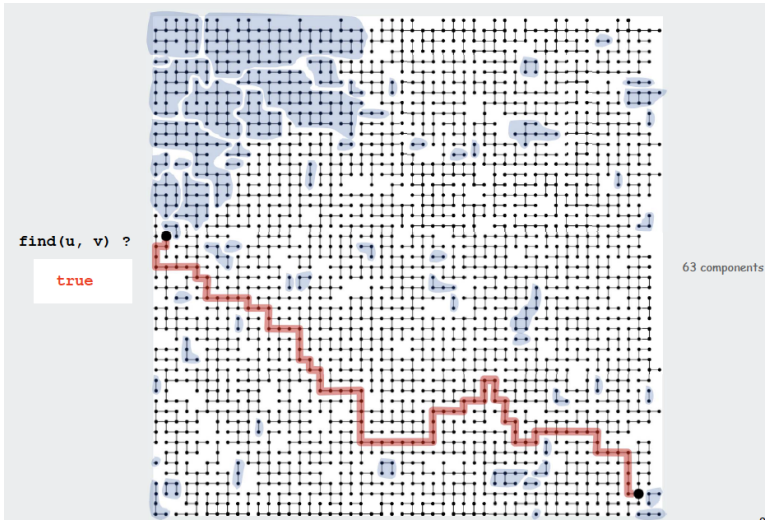


Extrait de *Algorithms* R. Sedgewick & K. Wayne, Addison Weysley (2011)

LE CONTEXTE

Réseau, problème de connexion

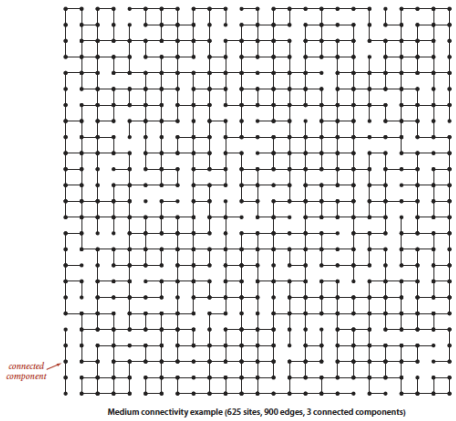


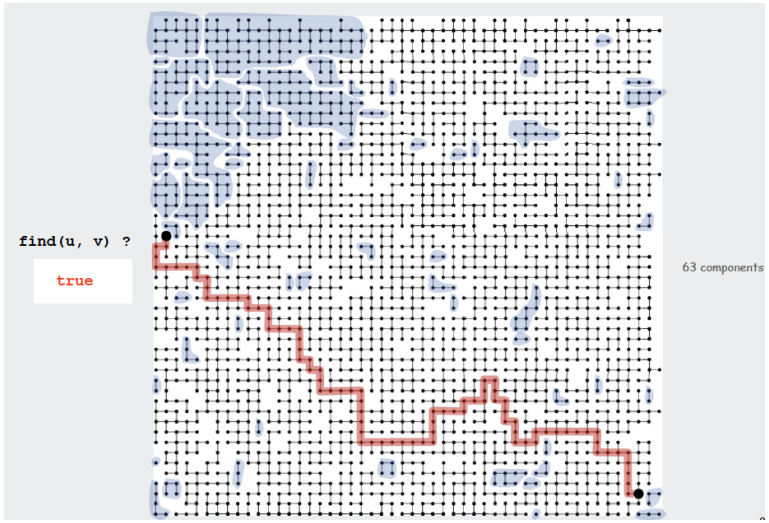


Extrait de *Algorithms* R. Sedgewick & K. Wayne, Addison Weysley (2011)

Le contexte

Réseau, problème de connexion





Extrait de *Algorithms* R. Sedgewick & K. Wayne, Addison Weysley (2011)

Exemples

- réseaux d'ordinateurs
- pages web sur internet
- réseaux sociaux
- propagation d'épidémie :(
- traitement d'image (pixels dans une photo)
- percolation

UGA Université Grenoble Alpes

Le problème

Notations

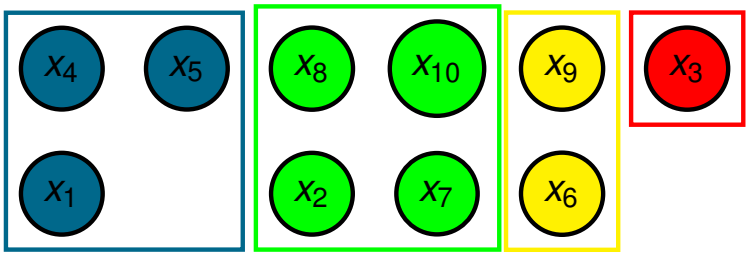
- E ensemble de n objets

  $$E = \{x_1, \cdots, x_{10}\}$$

- Une partition $\mathcal{P}$ de E est un ensemble de parties disjointes de E dont l'union est E

  $$\mathcal{P} = \{\{x_1, x_4, x_5\}, \{x_3\}, \{x_2, x_7, x_8, x_{10}\}, \{x_6, x_9\}\}$$

  Partition de E en 4 sous-ensembles



Le problème

Notations

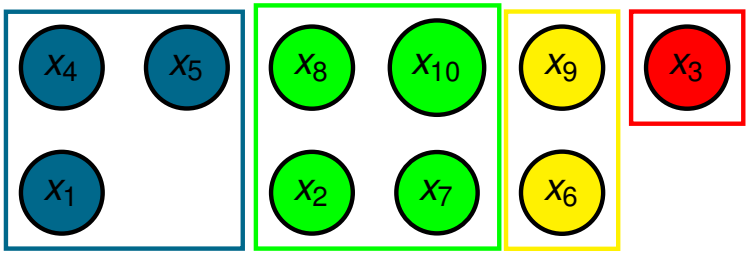
- E ensemble de n objets

  $$E = \{x_1, \cdots, x_{10}\}$$

- Une partition $\mathscr{P}$ de E est un ensemble de parties disjointes de E dont l'union est E

  $$\mathscr{P} = \{\{x_1, x_4, x_5\}, \{x_3\}, \{x_2, x_7, x_8, x_{10}\}, \{x_6, x_9\}\}$$

  Partition de E en 4 sous-ensembles



Opérations

- `Même_partie`(x_i, x_j) teste si x_i et x_j sont dans la même partie
  `Même_partie`(x_2, x_8) renvoie `vrai`
  `Même_partie`(x_1, x_9) renvoie `faux`
- `Union`(x_i, x_j) réunit les parties auxquelles x_i et x_j appartiennent
  `Union`(x_4, x_6) transforme $\mathscr{P}$ en $\{\{x_1, x_4, x_5, x_6, x_9\}, \{x_3\}, \{x_2, x_7, x_8, x_{10}\}, \}$

UGA Université Grenoble Alpes

EXERCICE1

Brainstorming 1

Proposer une structure de donnée pour représenter une partition

EXERCICE1

Brainstorming 1

Proposer une structure de donnée pour représenter une partition

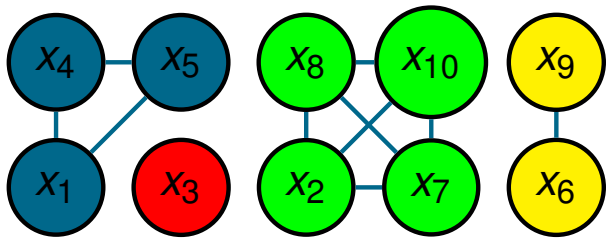
Premiers algorithmes

En considérant une structure de graphe représentée par sa matrice d'adjacence A

Matrice

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Représentation



- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

EXERCICE 2

Brainstorming 2

Proposer une structure de donnée plus compacte pour représenter une partition.

Exercice 2

Brainstorming 2

Proposer une structure de donnée plus compacte pour représenter une partition.

Algorithme nouvelle version

Ne garder que les lignes représentant les parties

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

Exercice 2

Brainstorming 2

Proposer une structure de donnée plus compacte pour représenter une partition.

Algorithme nouvelle version

Ne garder que les lignes représentant les parties

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

Brainstorming 3

Proposer une structure de donnée encore plus compacte pour représenter une partition.

UGA Université Grenoble Alpes

Exercice 2

Brainstorming 2

Proposer une structure de donnée plus compacte pour représenter une partition.

Algorithme nouvelle version

Ne garder que les lignes représentant les parties

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

Brainstorming 3

Proposer une structure de donnée encore plus compacte pour représenter une partition.

Algorithme nouvelle nouvelle version

Vecteur contenant les identifiants des parties

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

UGA Université Grenoble Alpes

EXERCICE 3

Brainstorming 4

Changer de point de vue et considérer les parties contenant les éléments. Proposer une structure de donnée pour représenter une partition avec ce point de vue.

EXERCICE 3

Brainstorming 4

Changer de point de vue et considérer les parties contenant les éléments. Proposer une structure de donnée pour représenter une partition avec ce point de vue.

Algorithme nouvelle version

Une partie est une liste d'éléments

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

UGA Université Grenoble Alpes

Exercice 3

Brainstorming 4

Changer de point de vue et considérer les parties contenant les éléments. Proposer une structure de donnée pour représenter une partition avec ce point de vue.

Algorithme nouvelle version

Une partie est une liste d'éléments

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

Brainstorming 5

Faire un tableau avec le coût de chaque opérateur pour chacune des structures de données, faire l'analyse de ce tableau.

UGA Université Grenoble Alpes

Exercice 3

Brainstorming 4

Changer de point de vue et considérer les parties contenant les éléments. Proposer une structure de donnée pour représenter une partition avec ce point de vue.

Algorithme nouvelle version

Une partie est une liste d'éléments

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

Brainstorming 5

Faire un tableau avec le coût de chaque opérateur pour chacune des structures de données, faire l'analyse de ce tableau.

Une forêt à notre secour

Vecteur contenant les identifiants des parties

- Écrire les algorithmes correspondant à `Même_partie` et `Union`.
- Analyser le coût de ces algorithmes.

UGA Université Grenoble Alpes

SYNTHÈSE

Structure d'Union-Find

- algorithme naïf
- algorithme `Quick-Find`
- algorithme `Quick-Union`
- algorithme `Union-Find Forest`
- algorithme `Union-Find Forest` avec compression de chemin

Synthèse

Structure d'Union-Find

- algorithme naïf
- algorithme `Quick-Find`
- algorithme `Quick-Union`
- algorithme `Union-Find Forest`
- algorithme `Union-Find Forest` avec compression de chemin

Quelques idées issues de l'activité

- Démarche de construction d'une structure de donnée
- Séparation de la sémantique de l'implémentation
- Le coût dépend de l'usage des opérateurs
- Coût repose sur un compromis temps/espace (mémoire)
- La structure d'arbre offre souvent un compromis pour accéder (hors tables de hachage)
- Le coût peut être amorti par une stratégie d'anticipation

Take home message : le choix de la représentation demande du temps

UGA Université Grenoble Alpes

Nombre de partitions

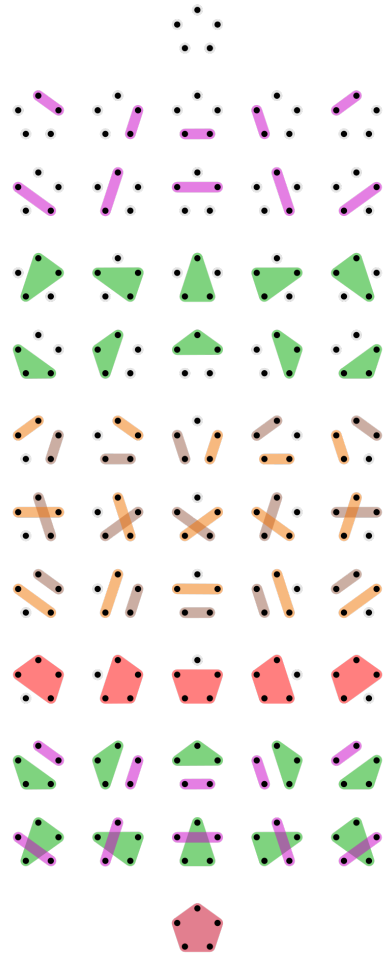
Nombres de Bell

n	0	1	2	3	4	5	$\cdots$
B_n	1	1	2	5	15	52	$\cdots$

Formule de récurrence : $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$

Dessiner les configurations correspondantes.
Pour plus d'informations
Site de l'OEIS (The On-line Encyclopedia of Integer Sequences

Pour $n = 5$



Wikipedia -By Watchduck (a.k.a. Tilman Piesk) - Own work, CC BY 3.0,