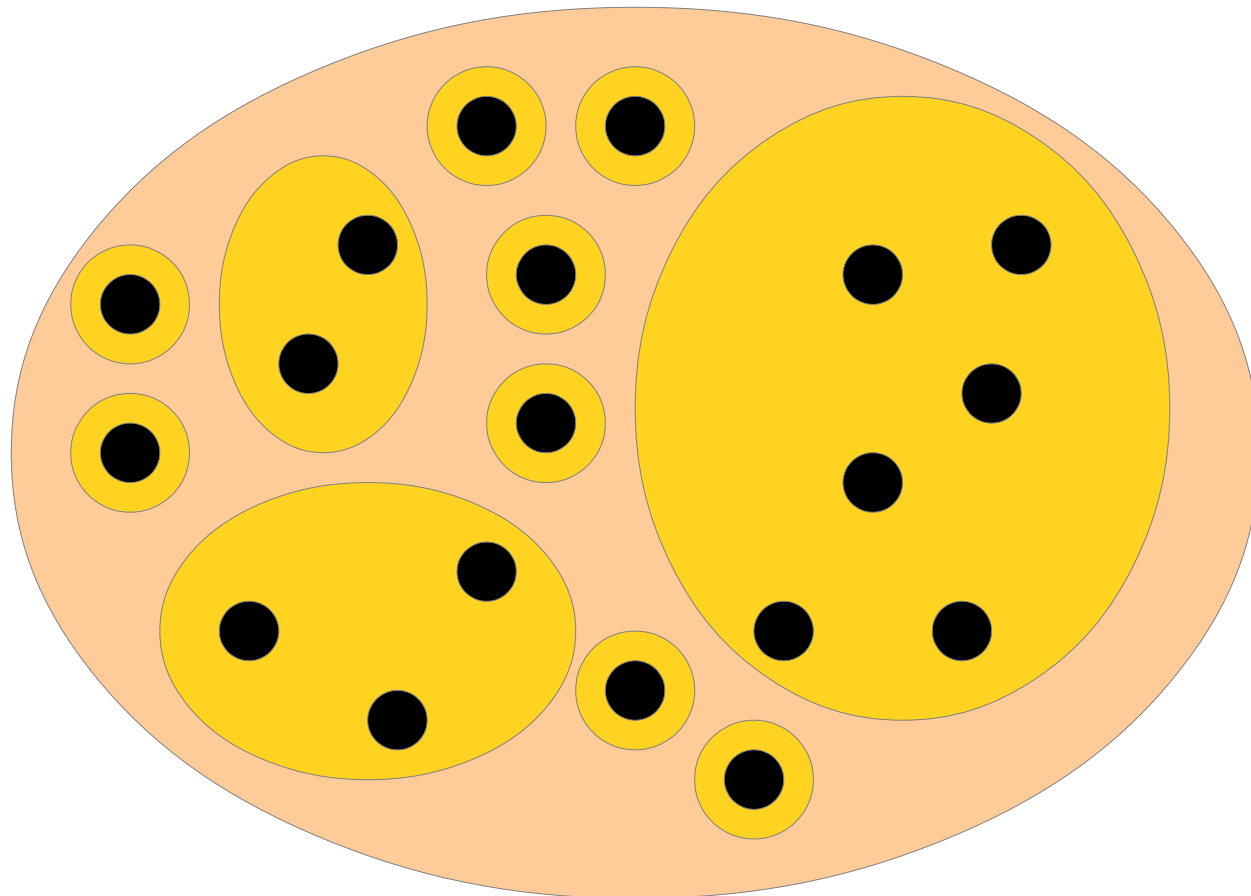
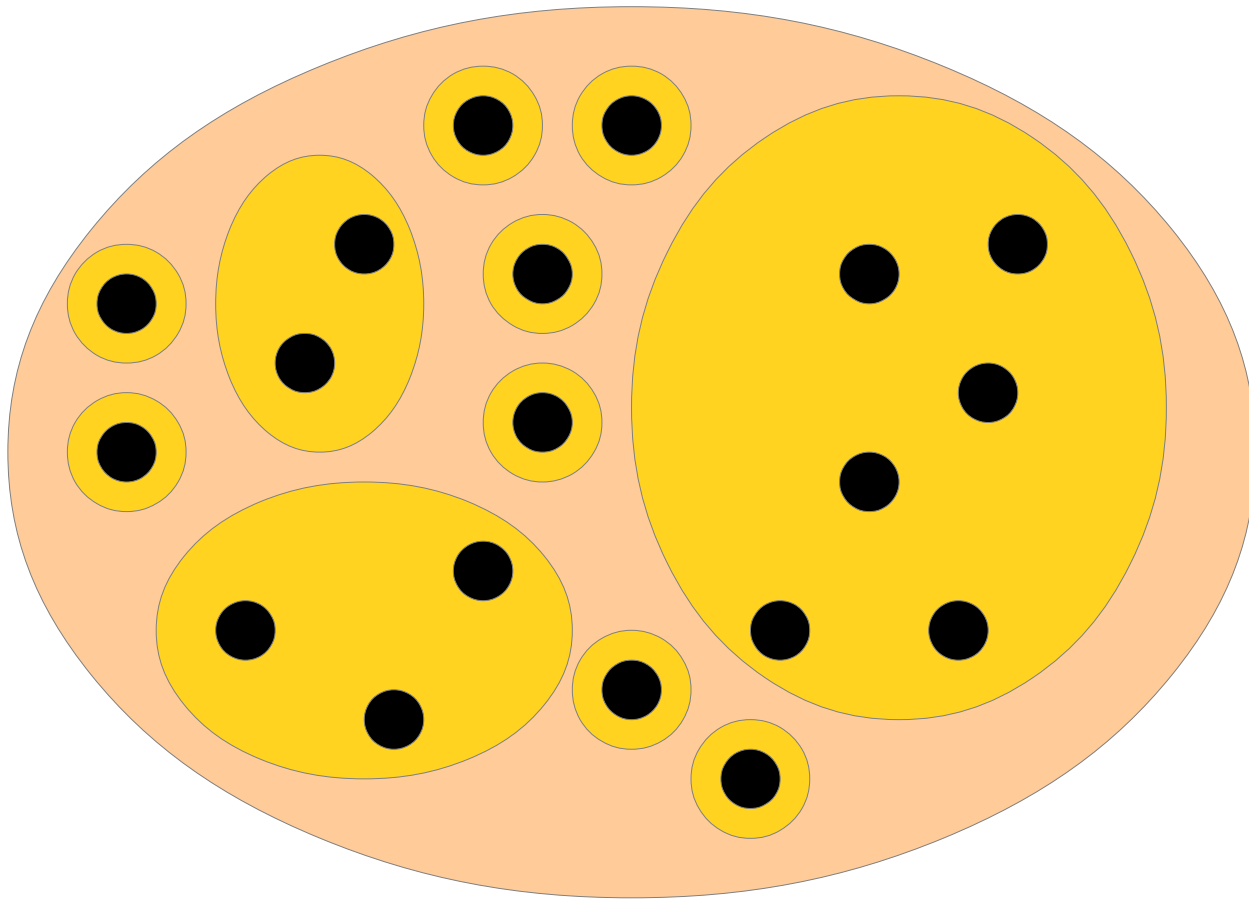


Relation d'équivalence



Relation d'équivalence

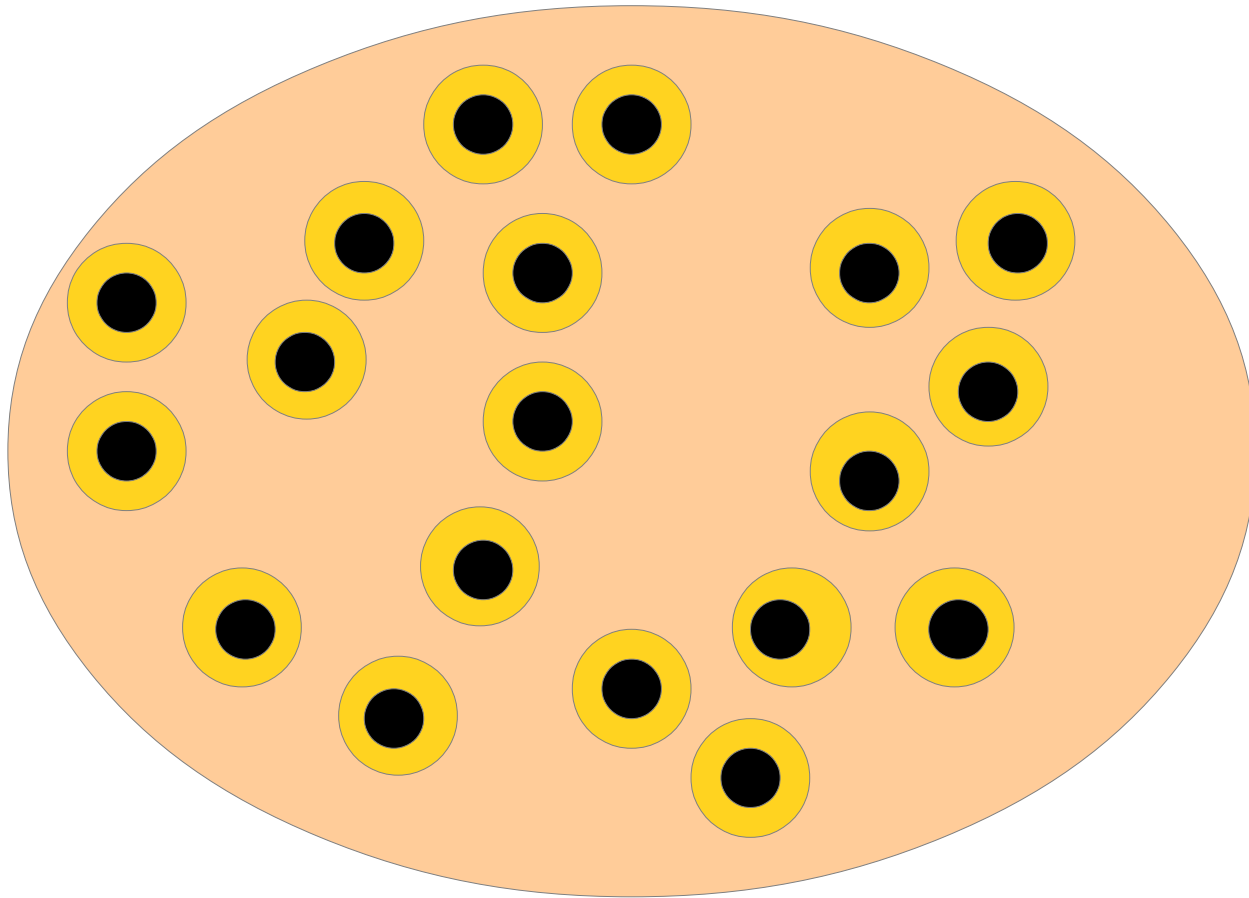


Créer une
partition triviale

Vérifier si $x \sim y$

Fusionner
deux classes

Relation d'équivalence

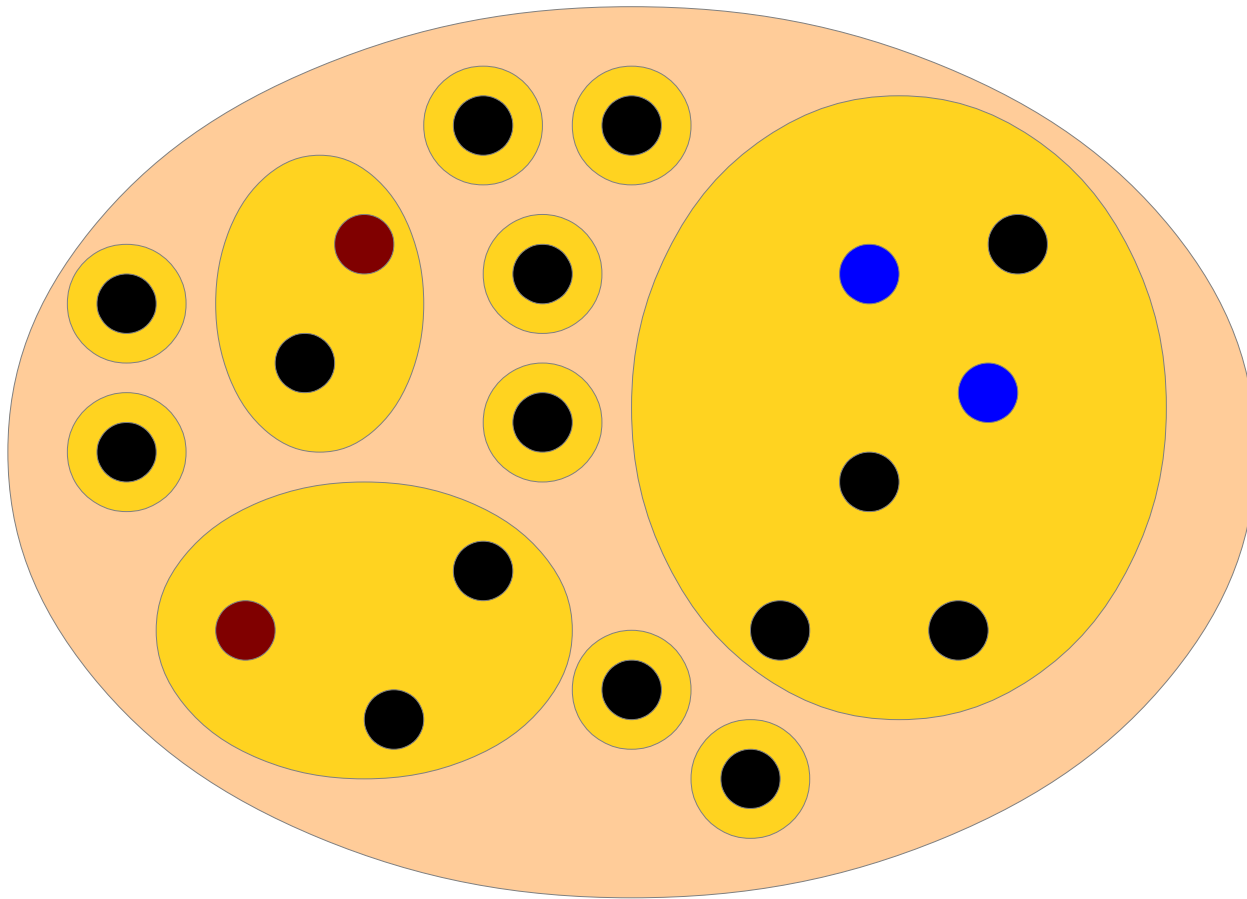


**Créer une
partition triviale**

Vérifier si $x \sim y$

Fusionner
deux classes

Relation d'équivalence

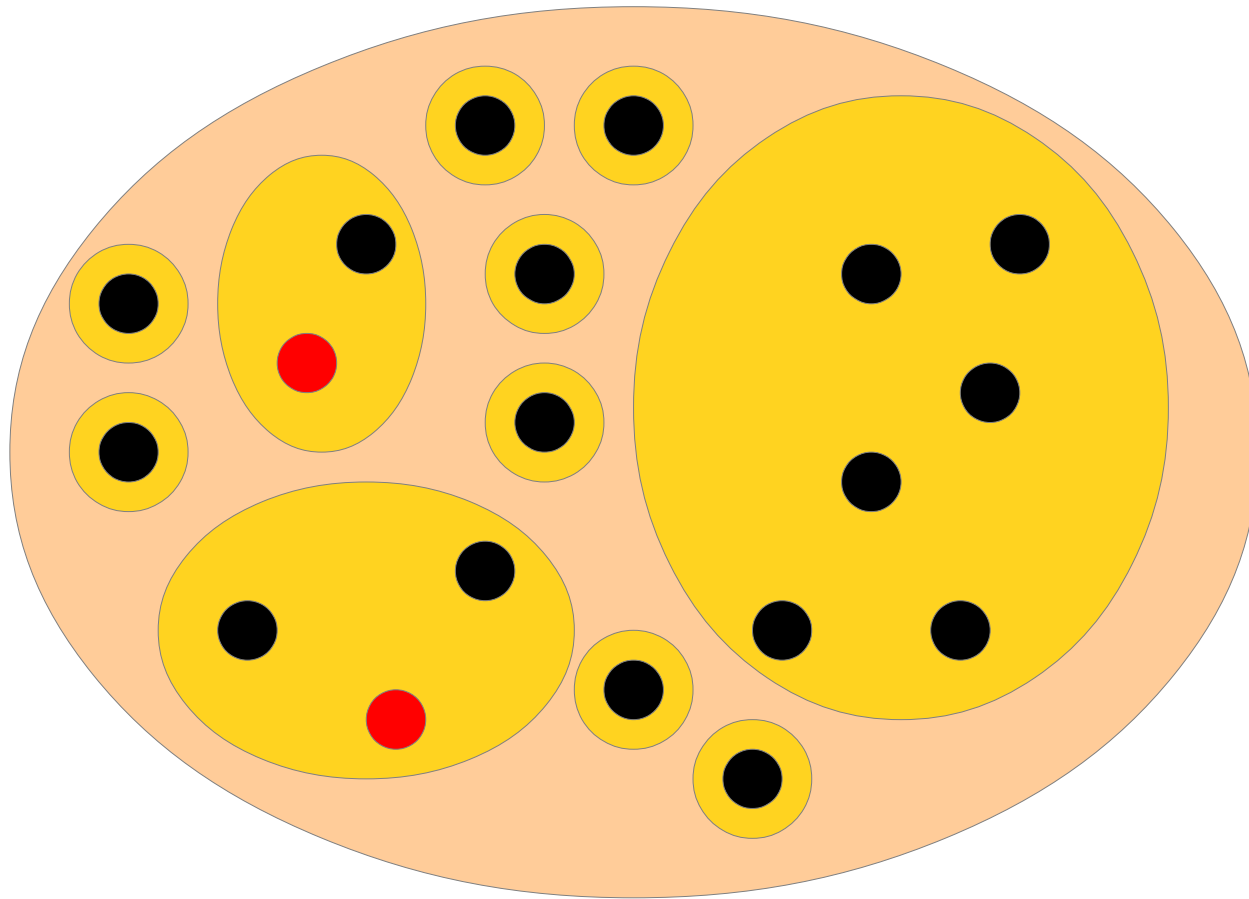


Créer une
partition triviale

Vérifier si $x \sim y$

Fusionner
deux classes

Union-find

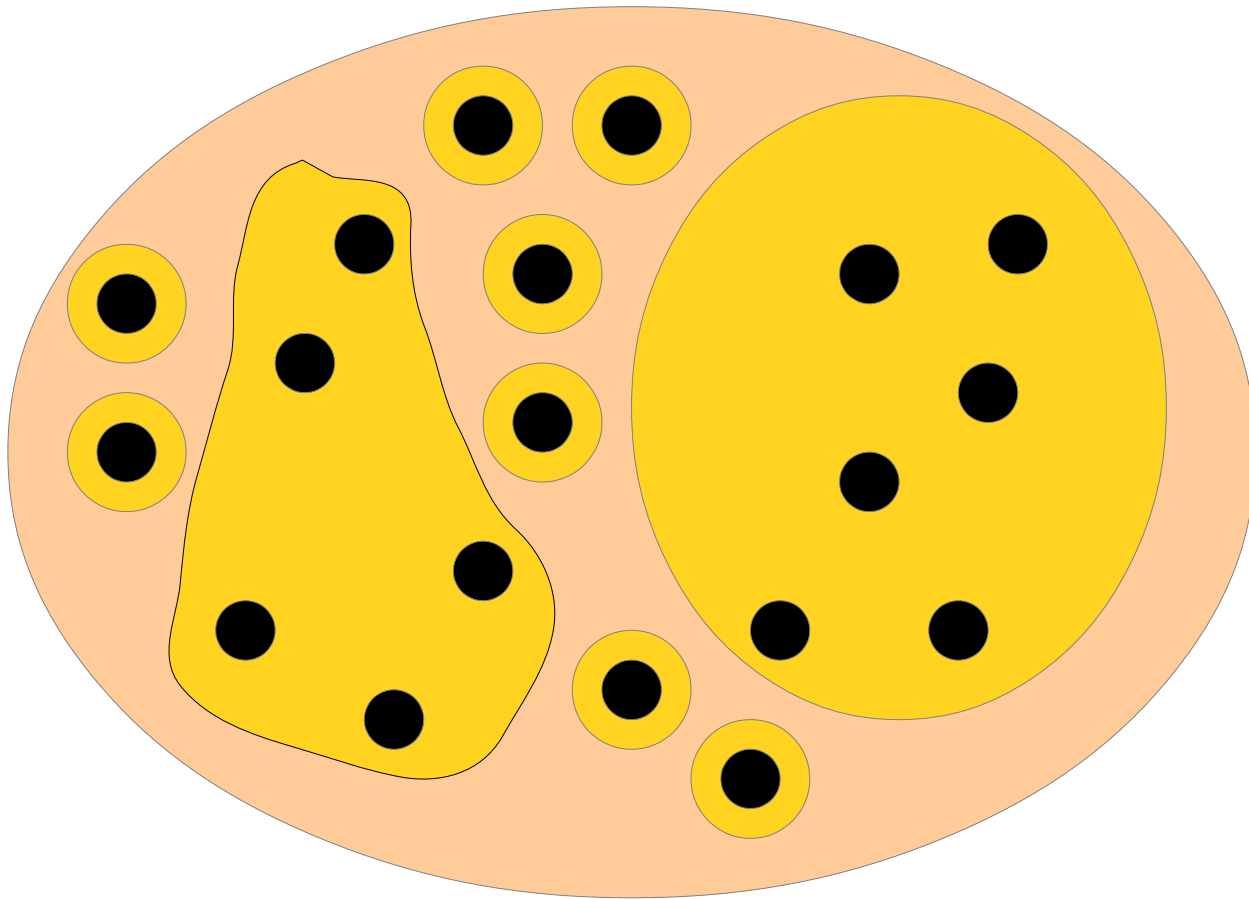


Créer une
partition triviale

Vérifier si $x \sim y$

**Fusionner
deux classes**

Relation d'équivalence

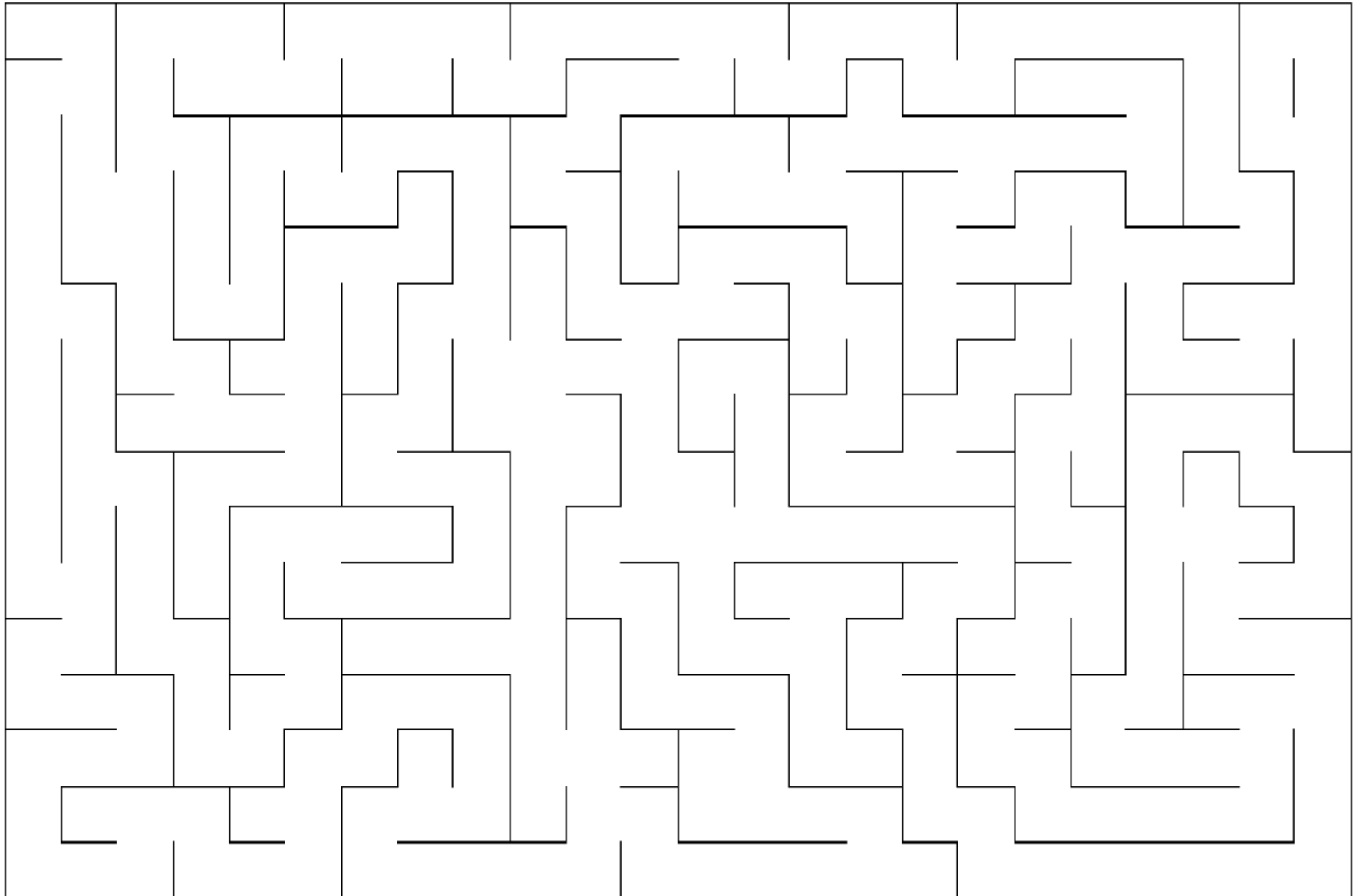


Créer une
partition triviale

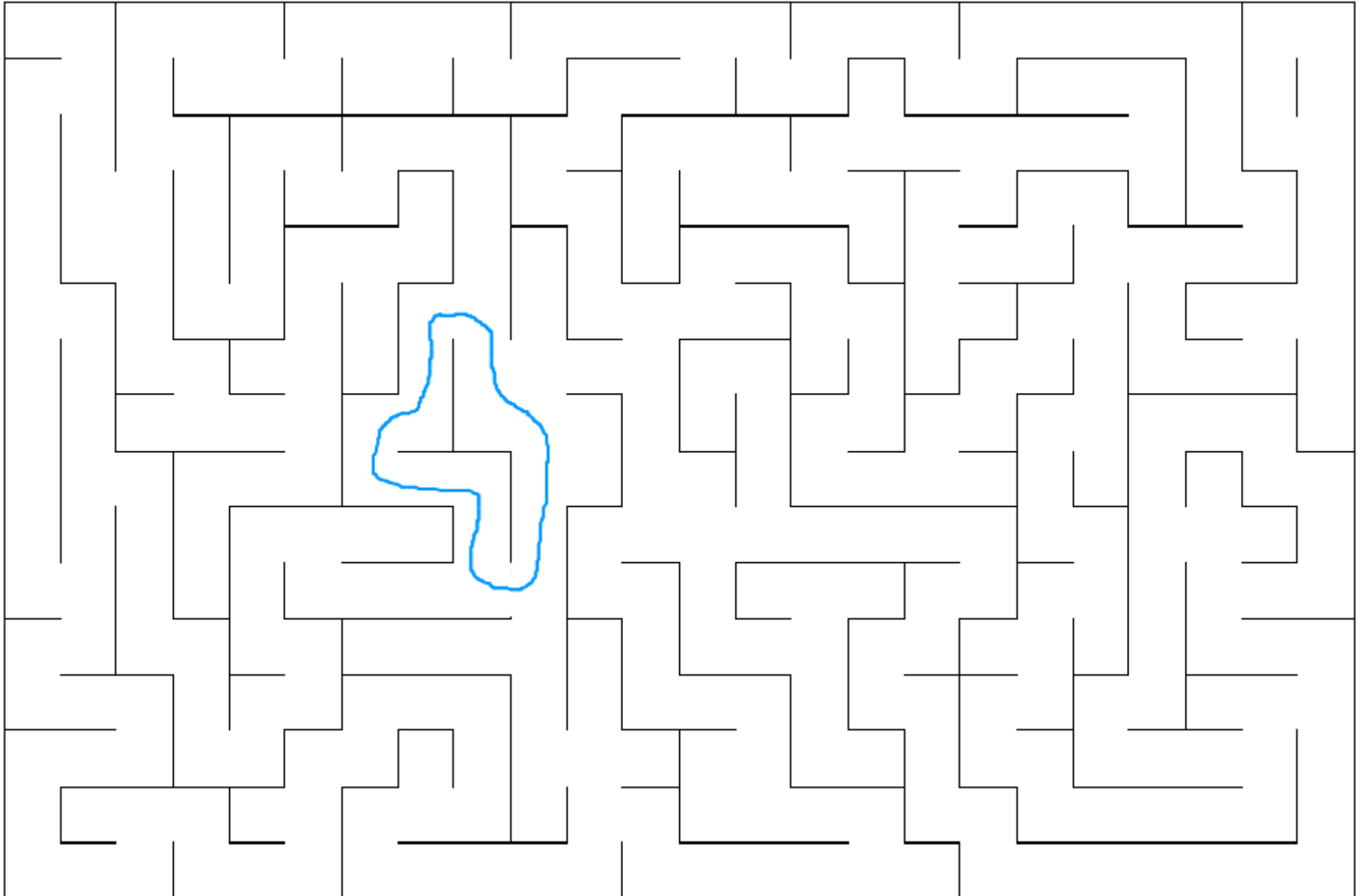
Vérifier si $x \sim y$

**Fusionner
deux classes**

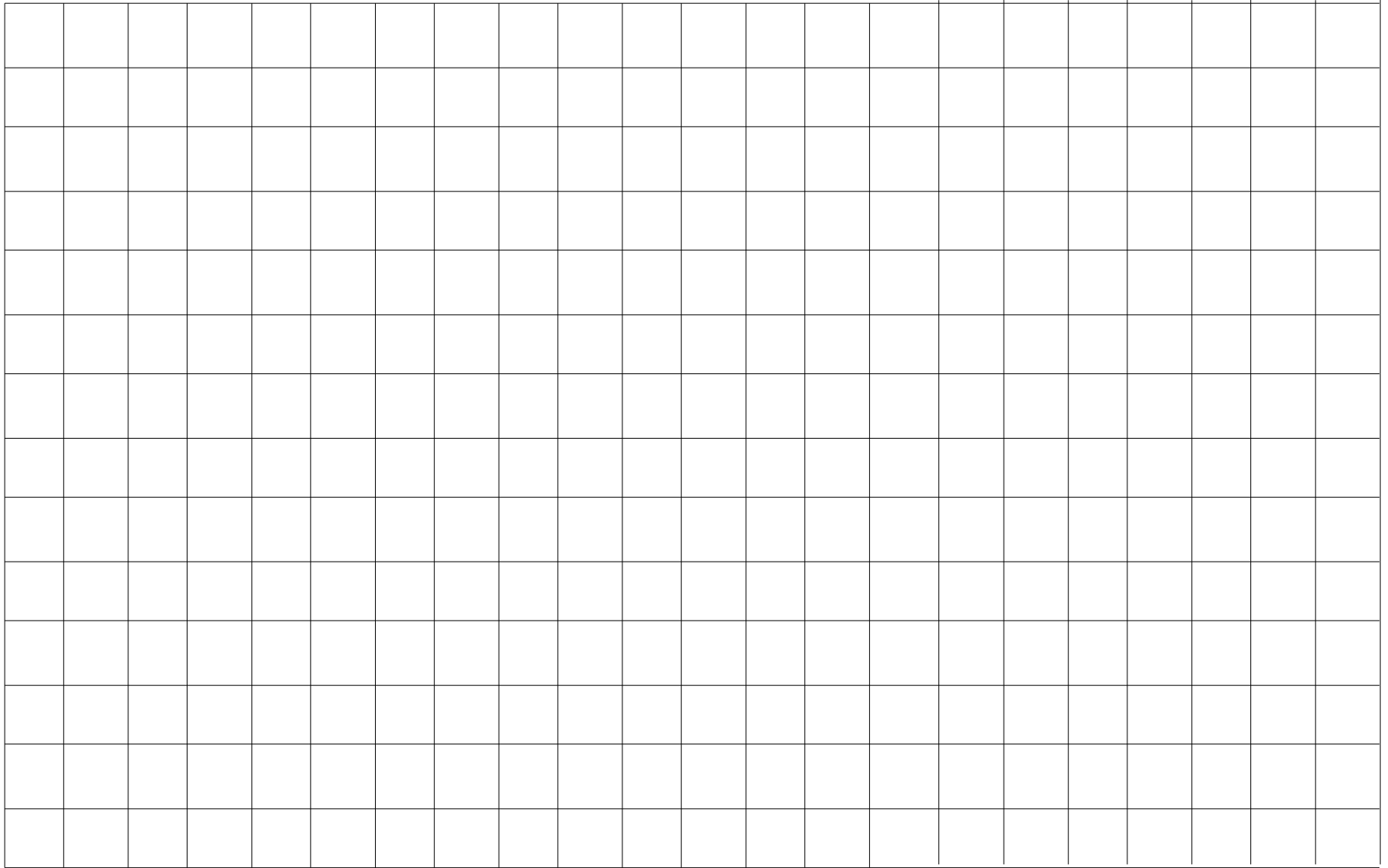
Motivation : générer un labyrinthe



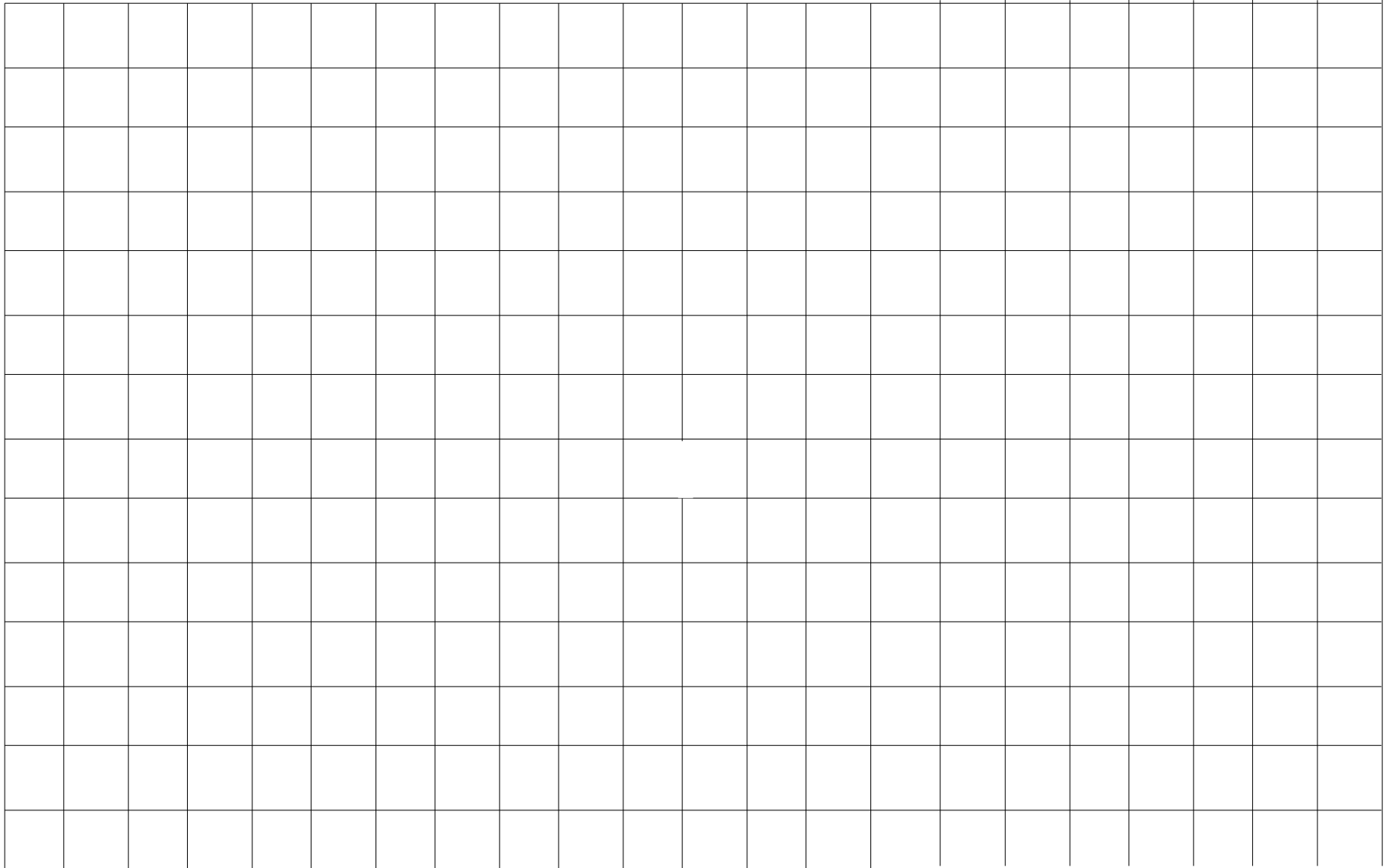
On ne veut pas de boucle



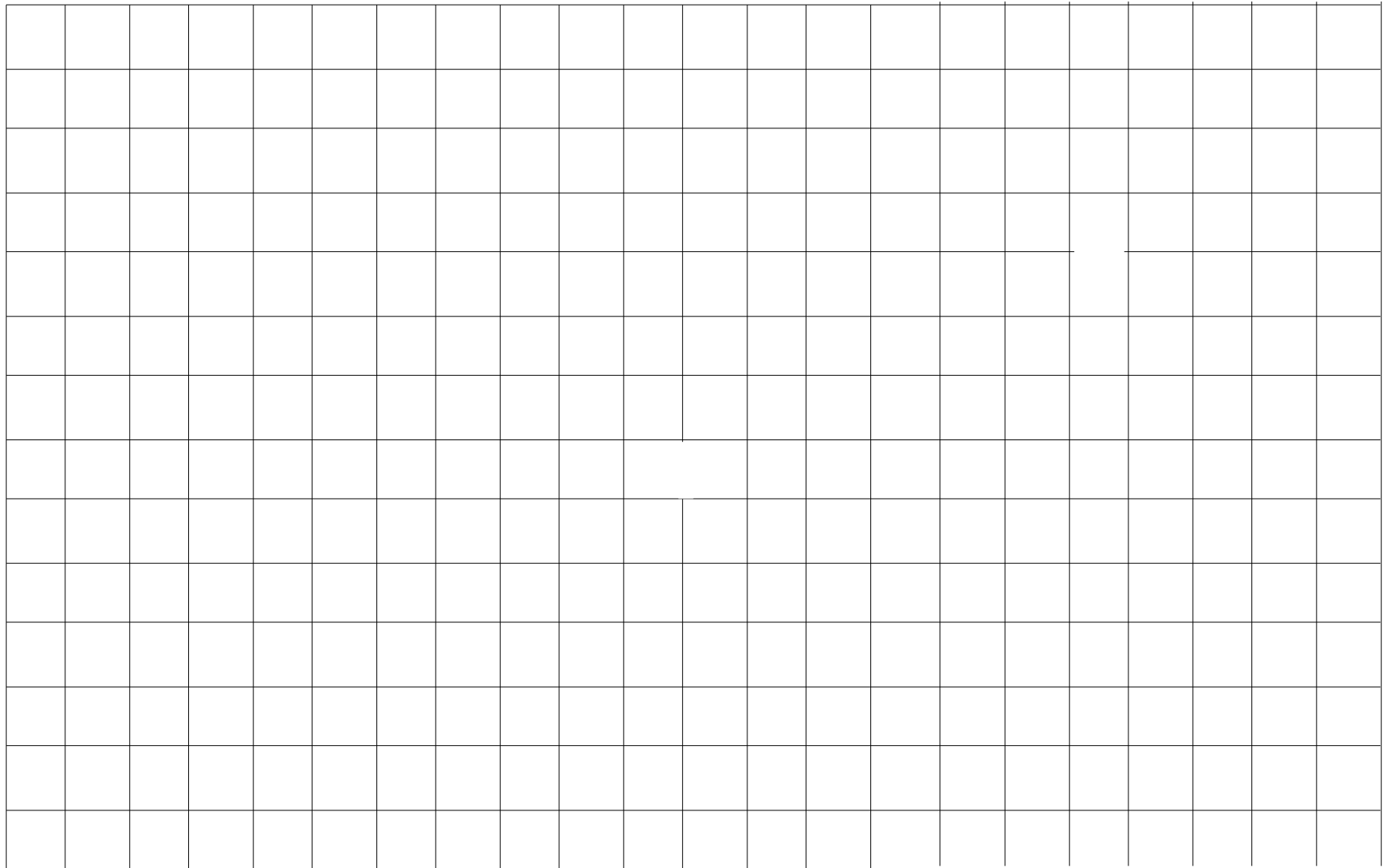
Démarche générale



Démarche générale



Démarche générale



Notre but

- On peut aller partout
- Pas de boucle

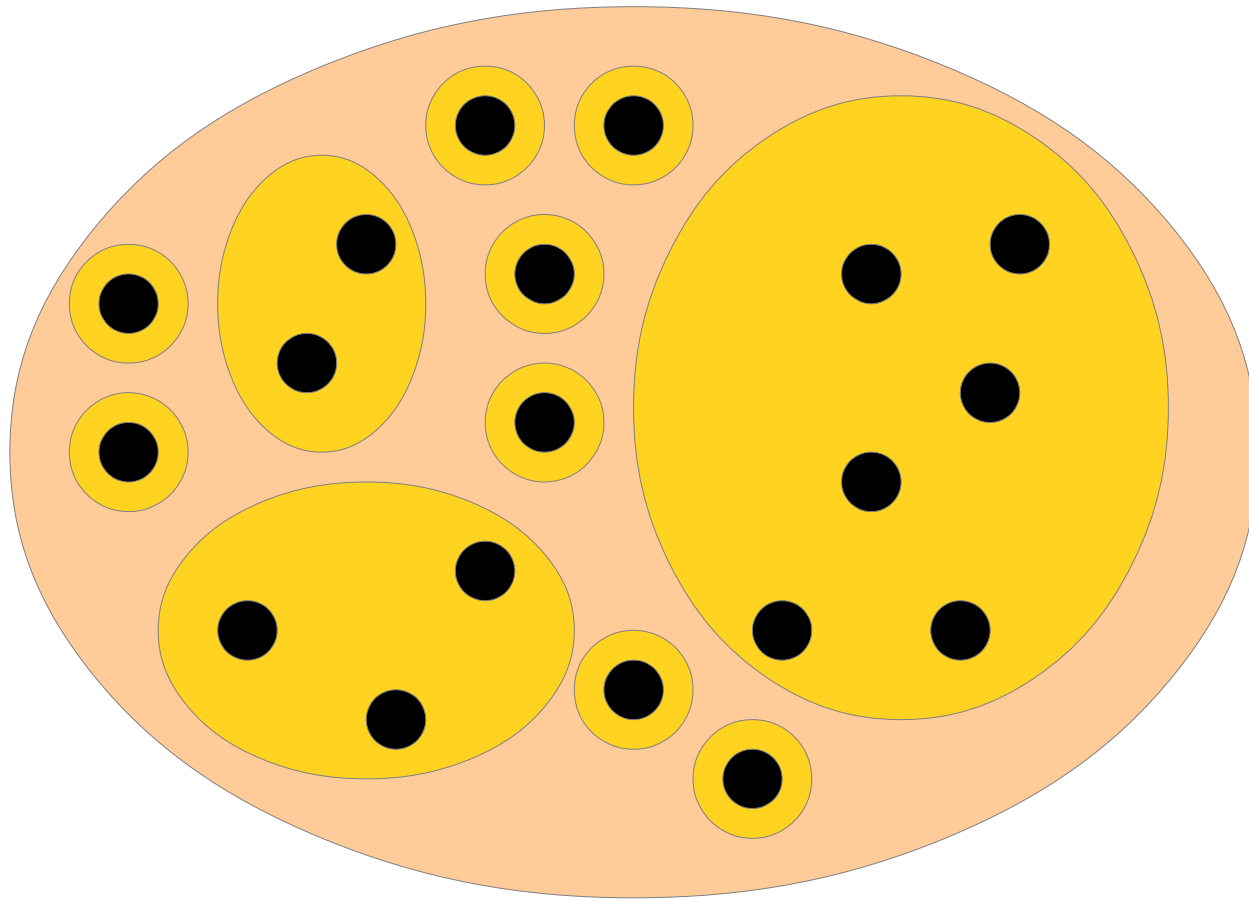
$c \sim d$: il y a un chemin de c à d

- Une classe d'équivalence
- On ne casse pas de mur entre c et d si $c \sim d$
- Casser un mur entre c et d : c et d sont dans la même classe

Implémentation naïve

Première implémentation avec des arbres

Union-find

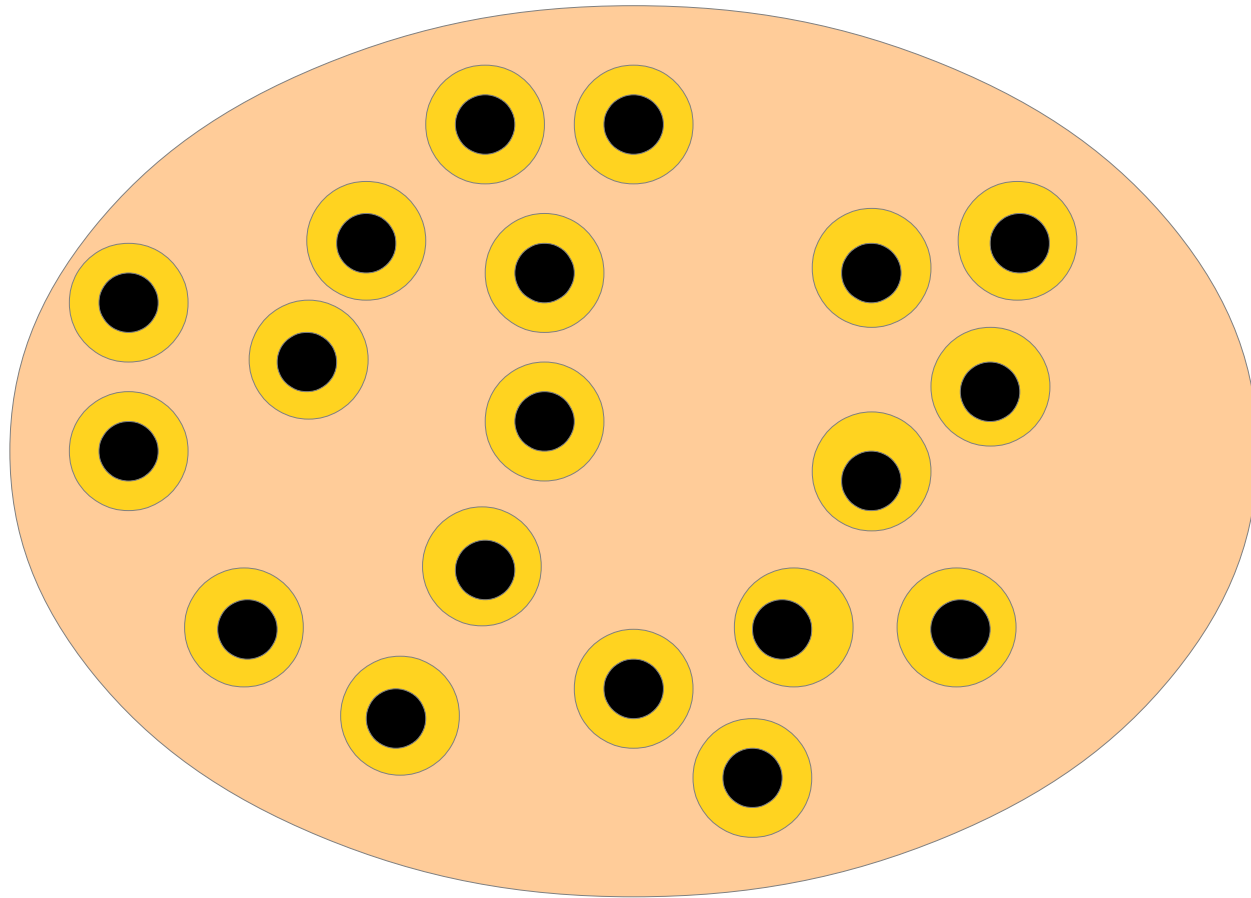


Créer union-find

Find

Union

Union-find

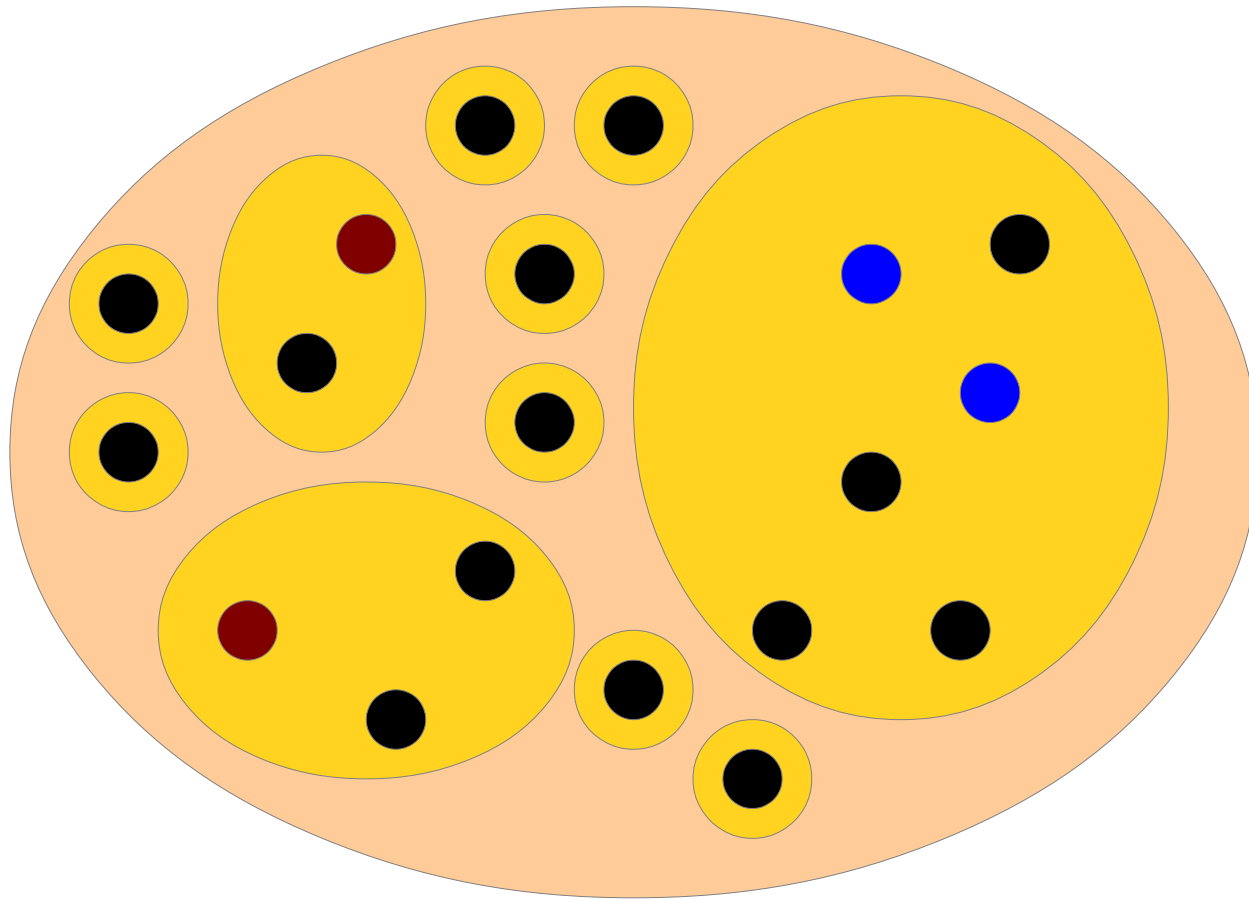


Créer union-find

Find

Union

Union-find

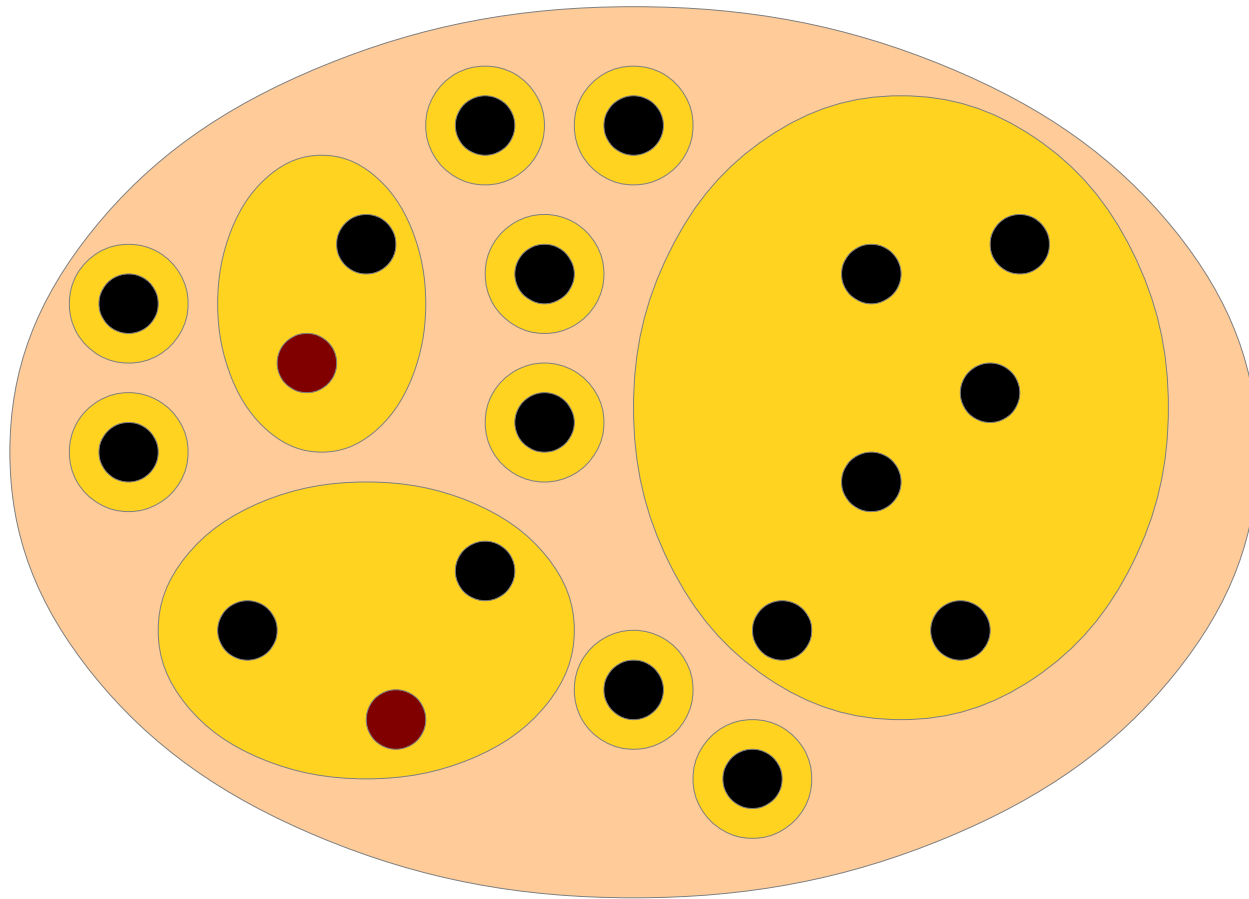


Créer union-find

Find

Union

Union-find

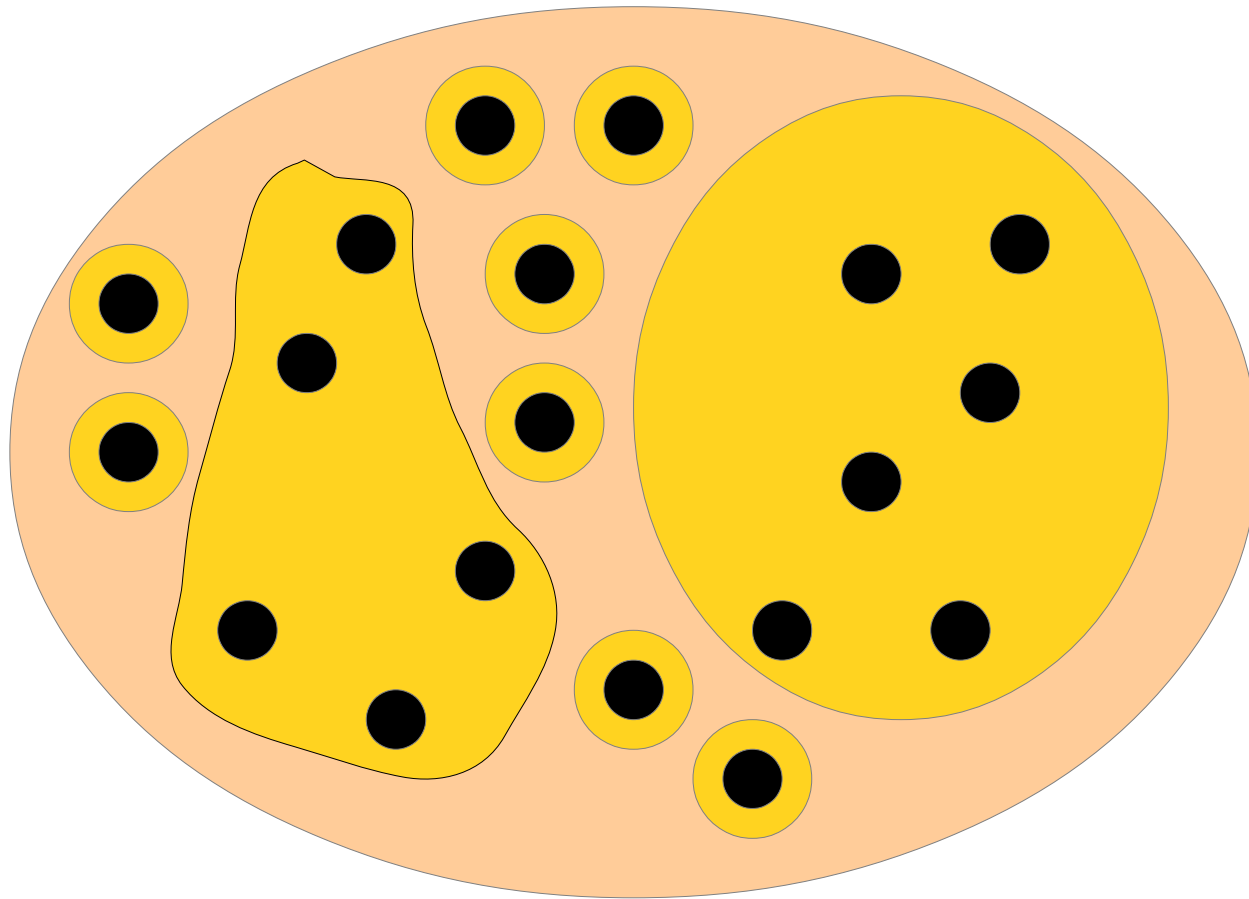


Créer union-find

Find

Union

Union-find

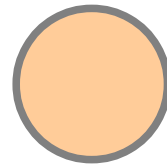
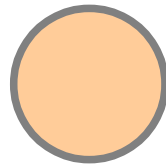
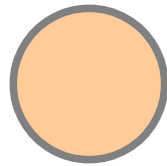
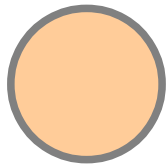
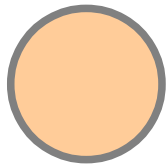
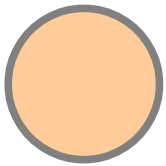


Créer union-find

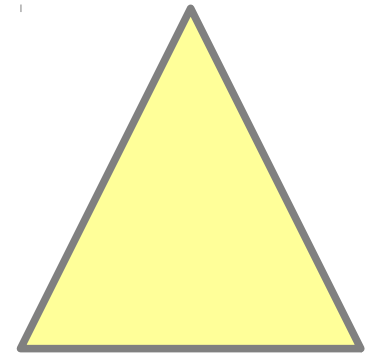
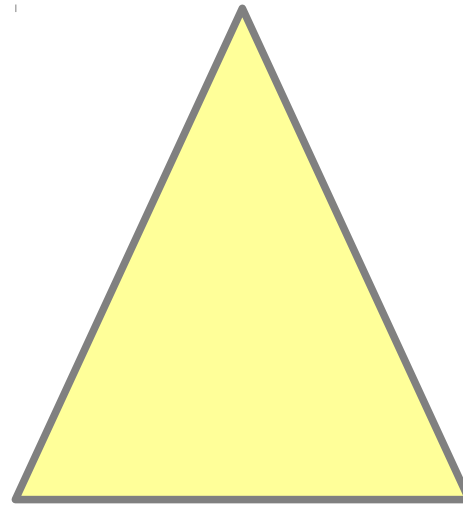
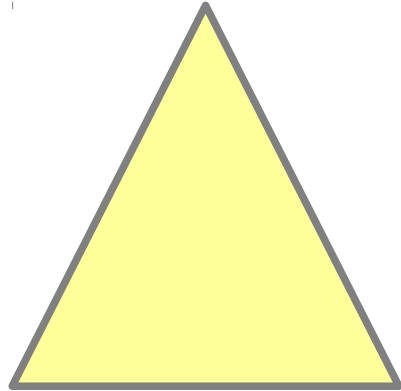
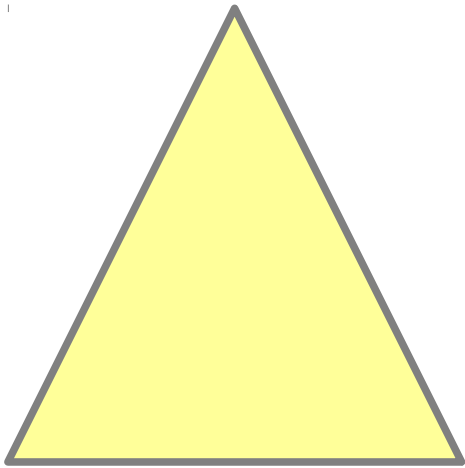
Find

Union

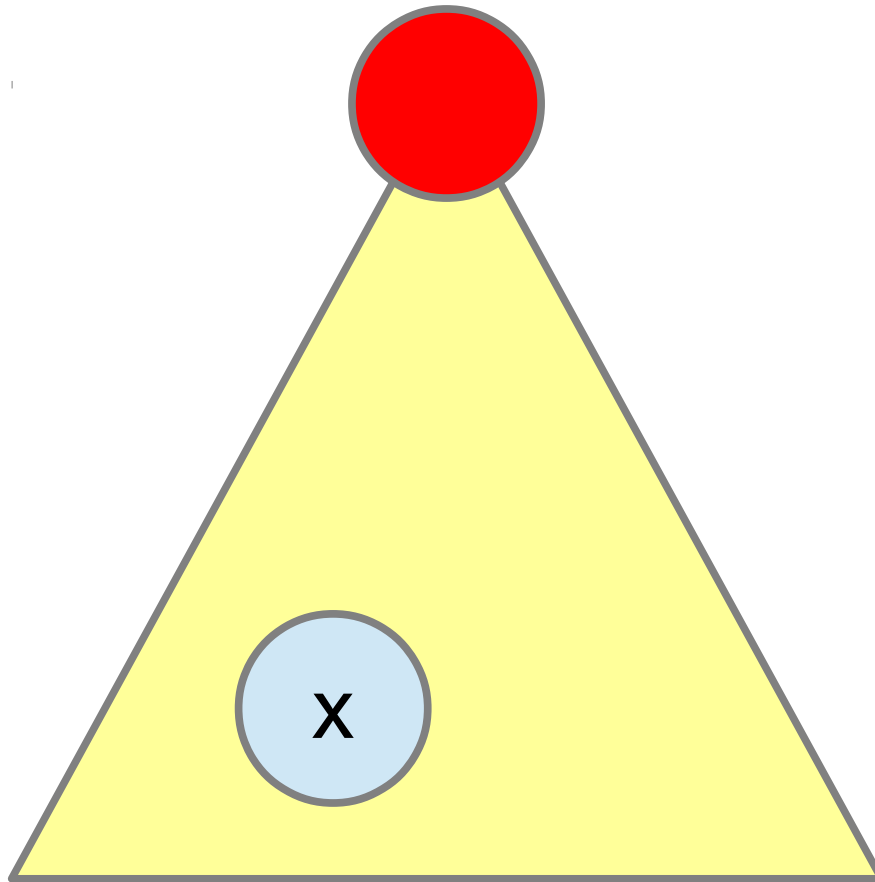
Une partition = une forêt



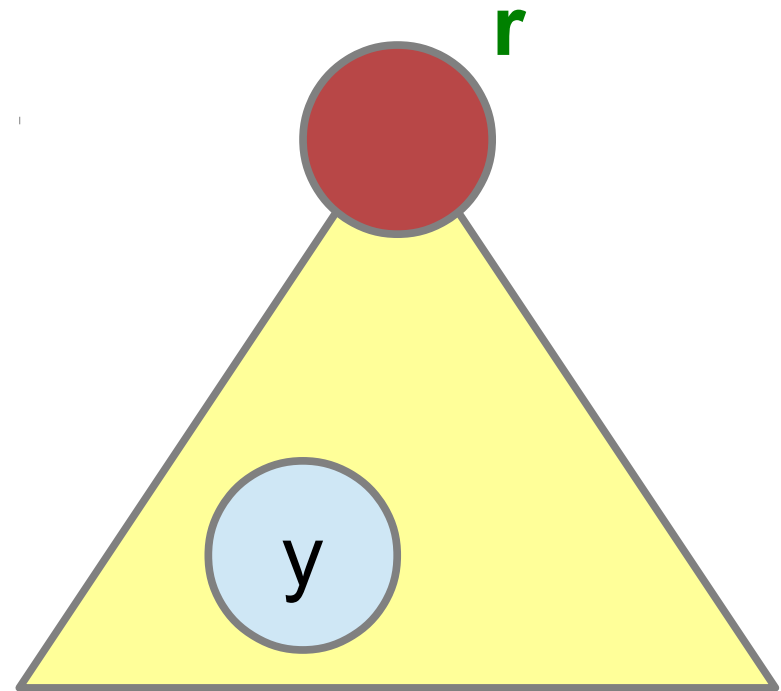
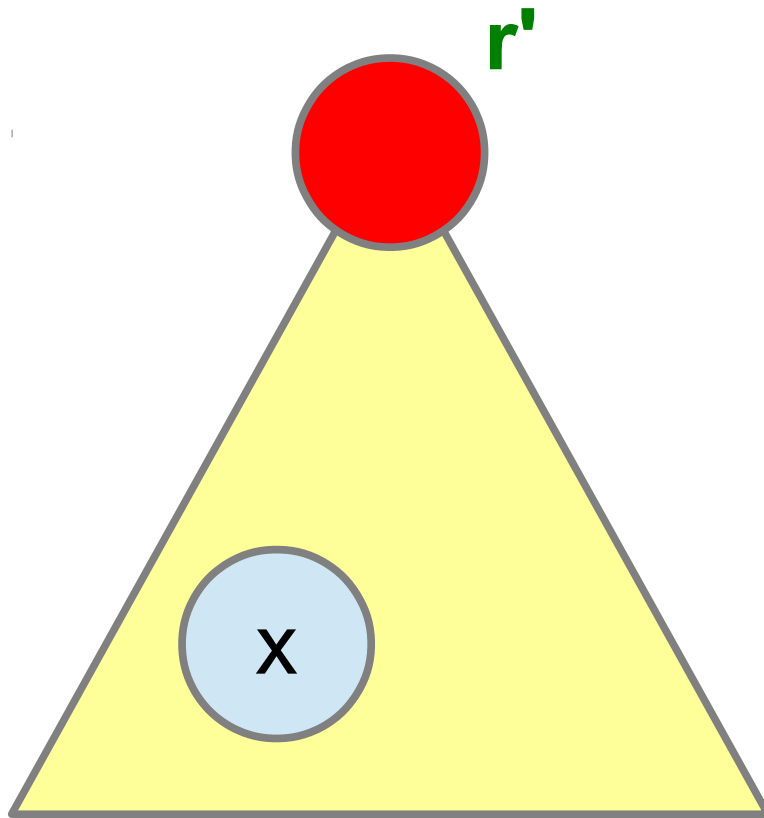
Une partition = une forêt



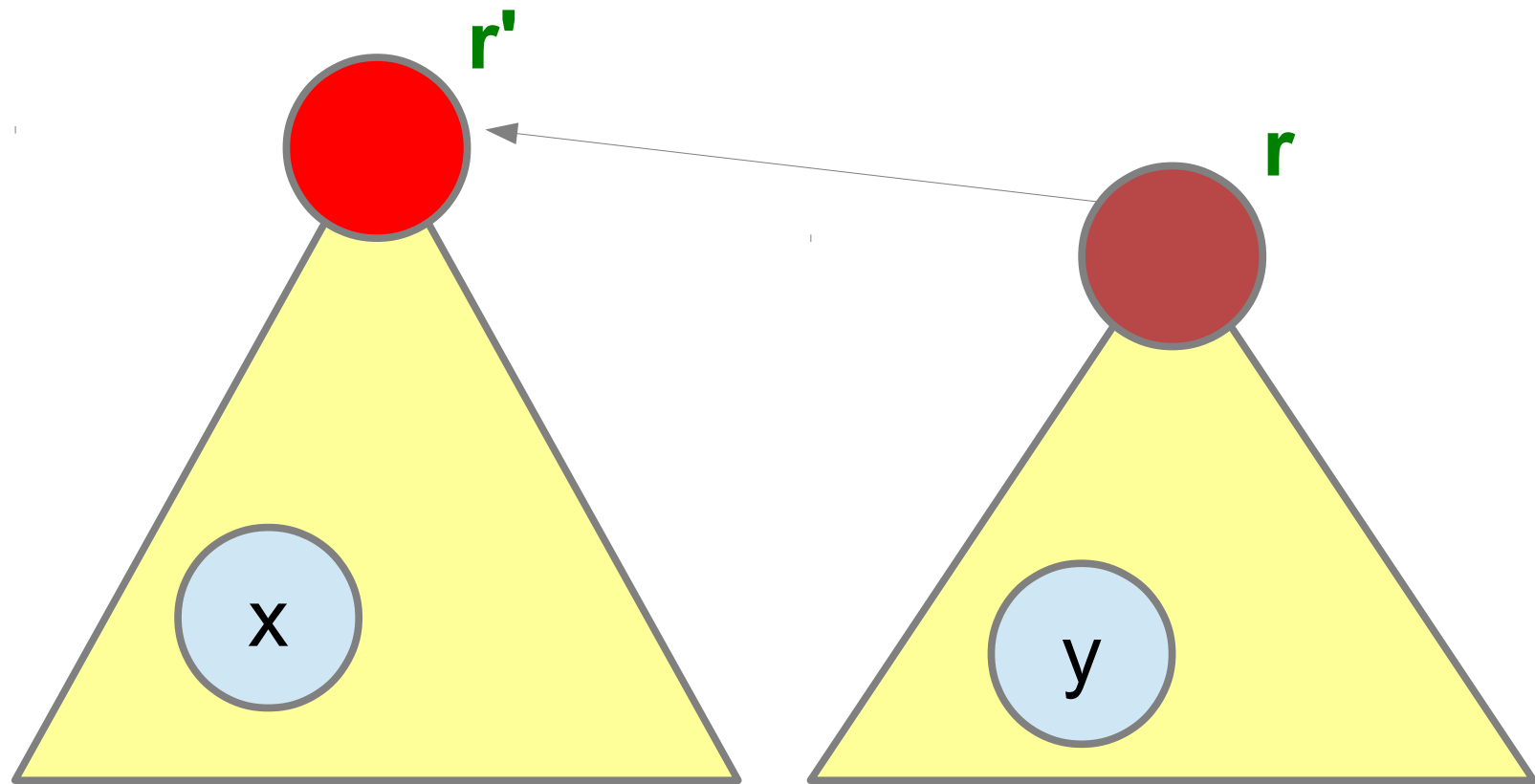
Find(x)



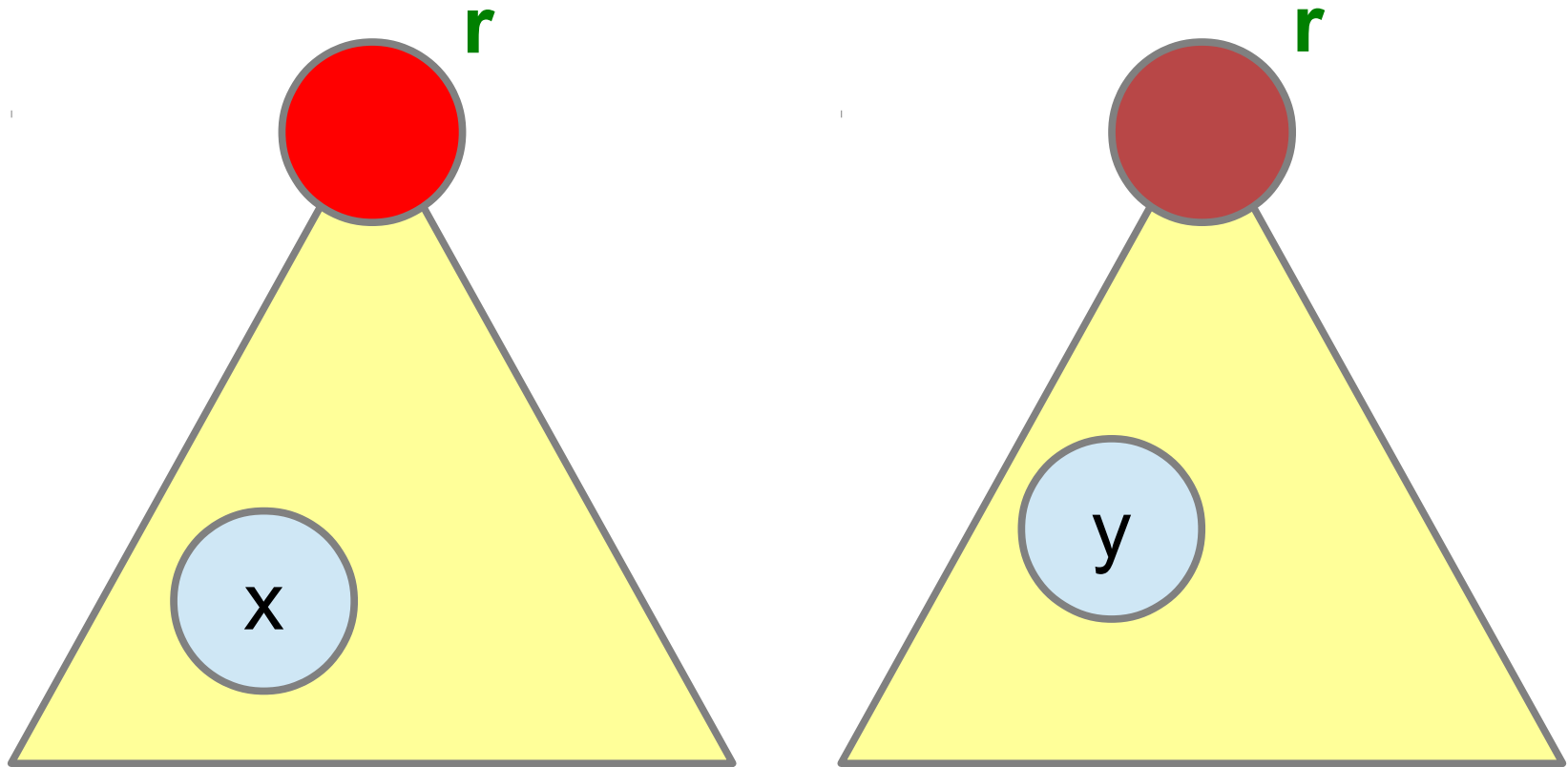
Union(x, y)



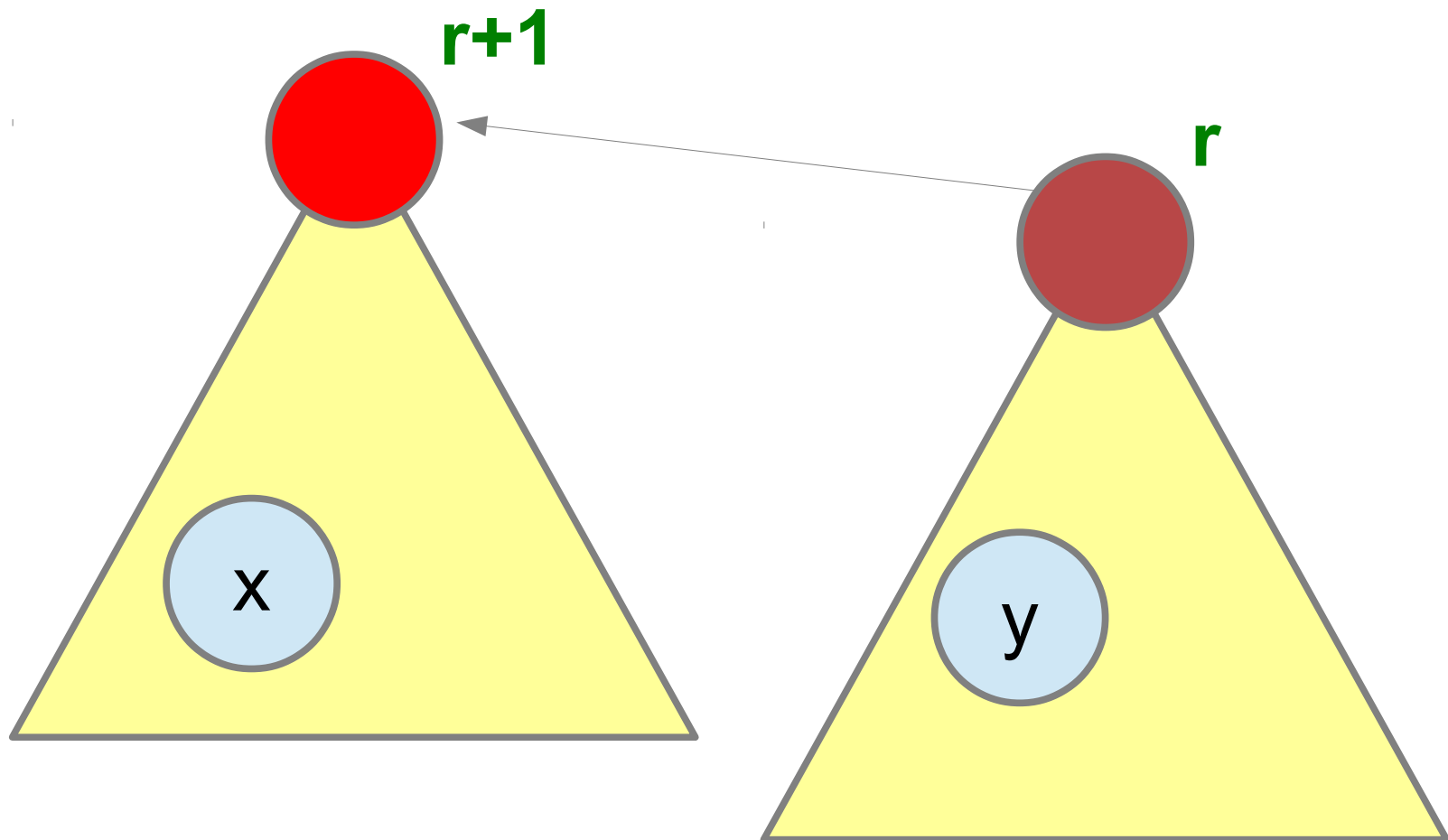
Union(x, y)



Union(x, y)

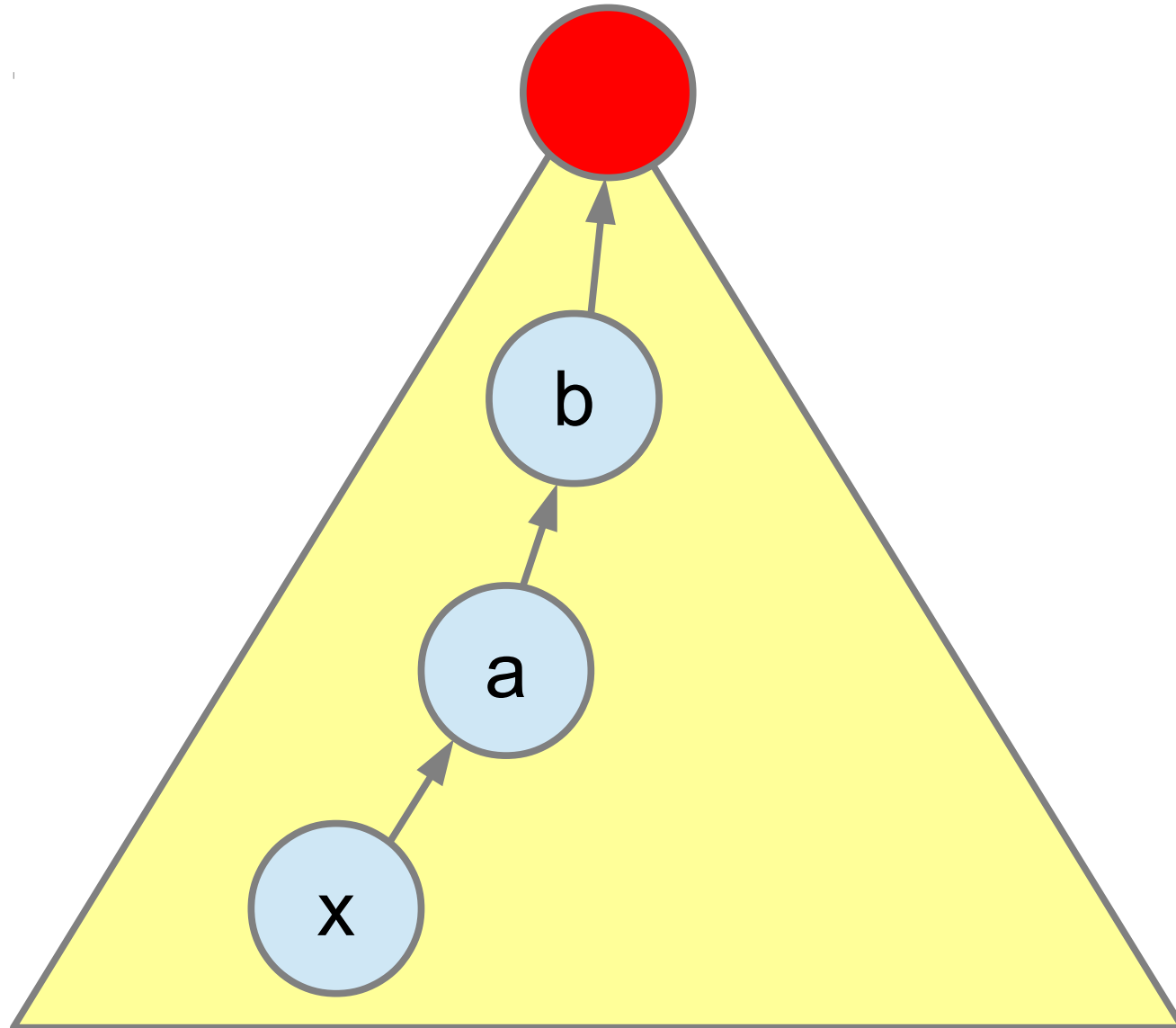


Union(x, y)

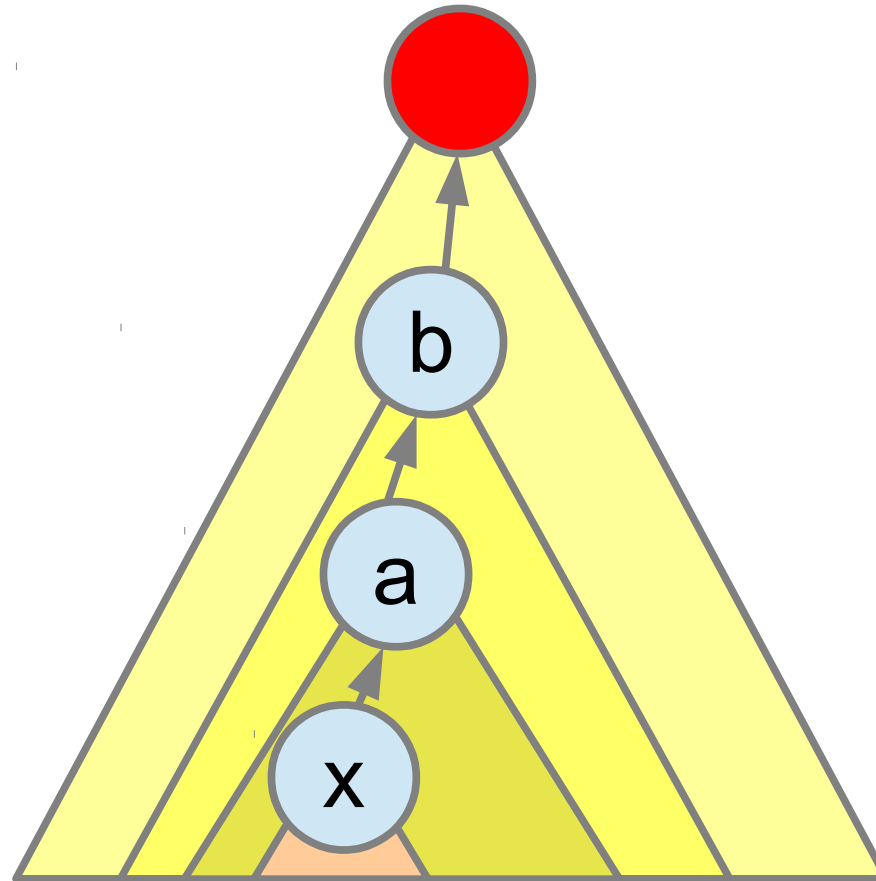


Amélioration de l'implémentation : compression de chemin

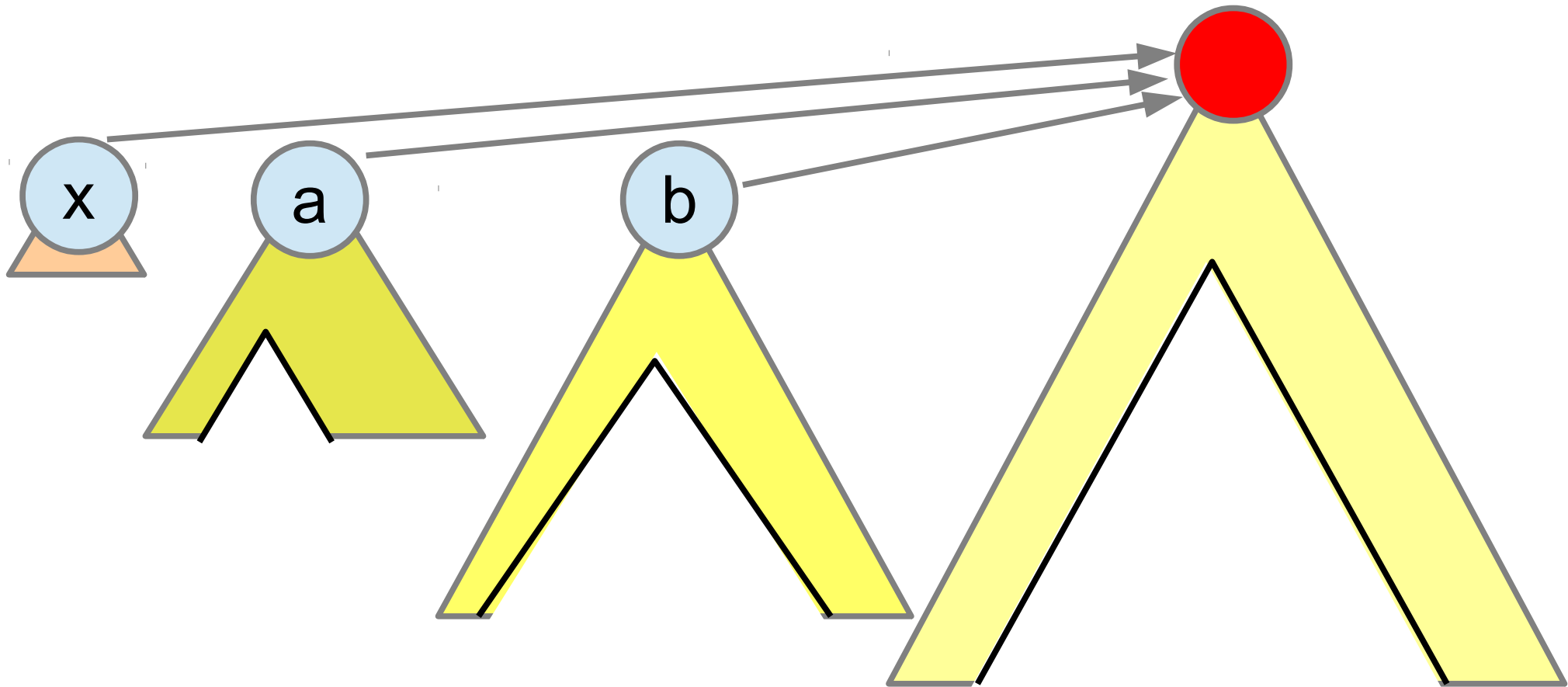
Find(x) avant



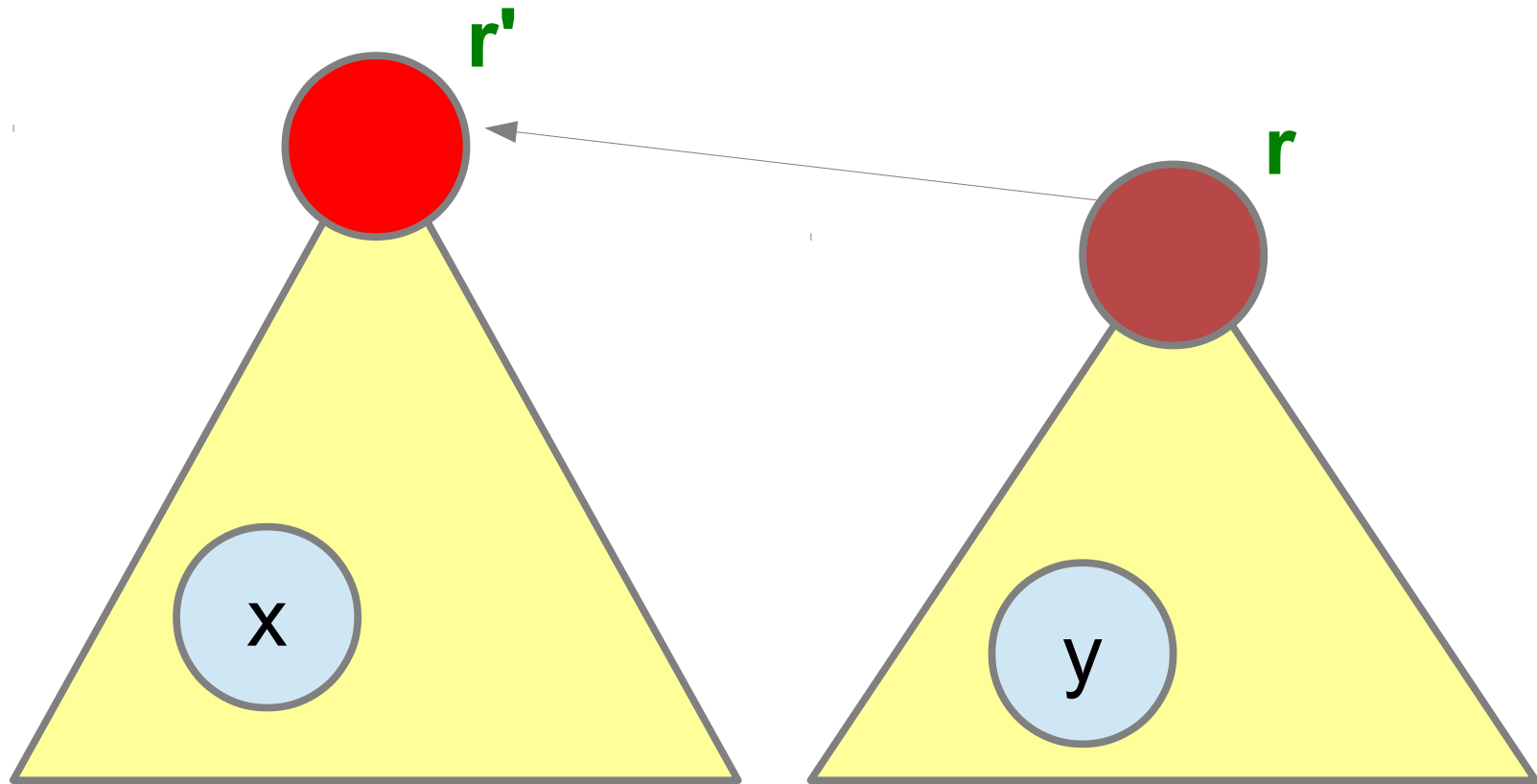
Find(x) avec compression de chemin



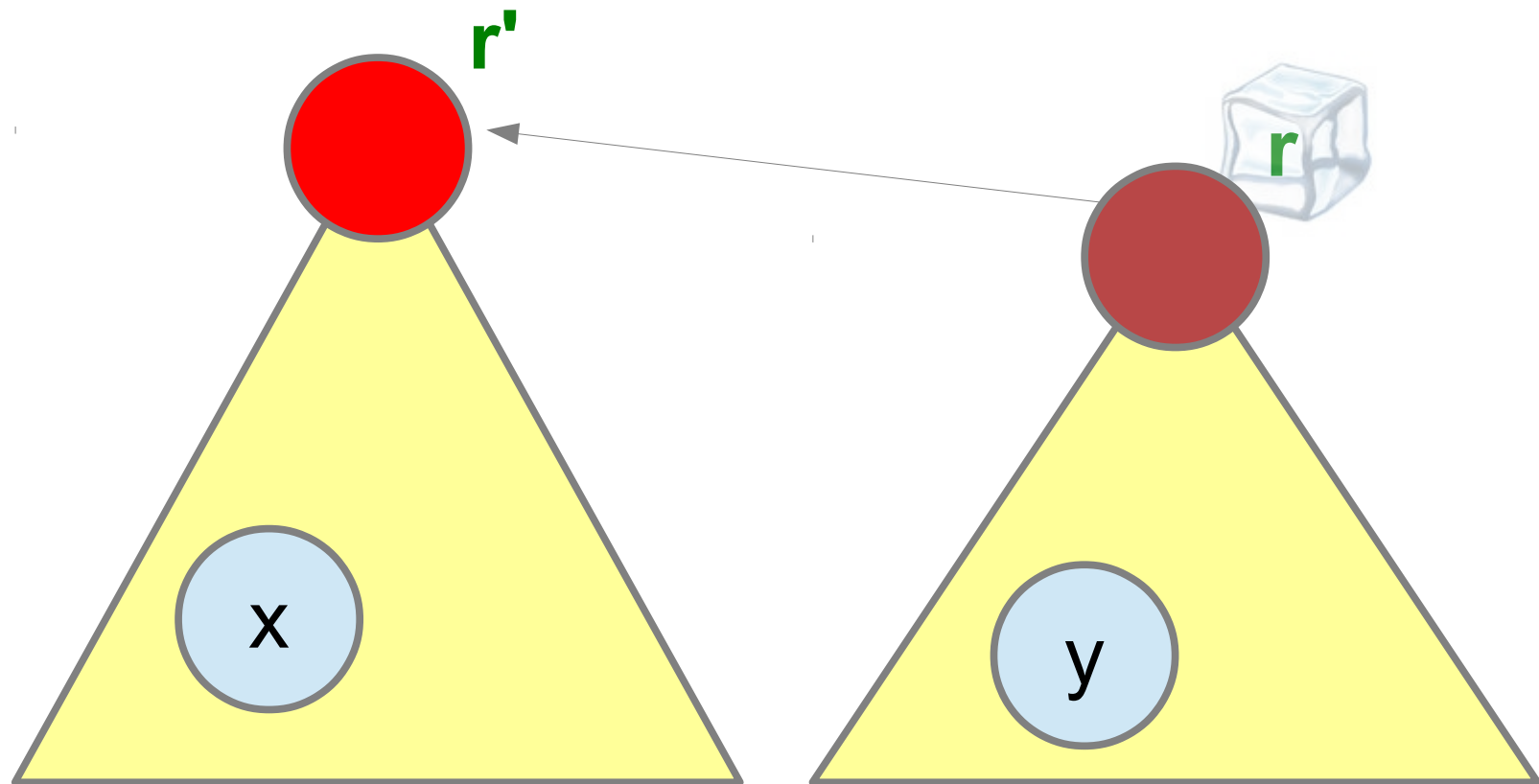
Find(x) avec compression de chemin



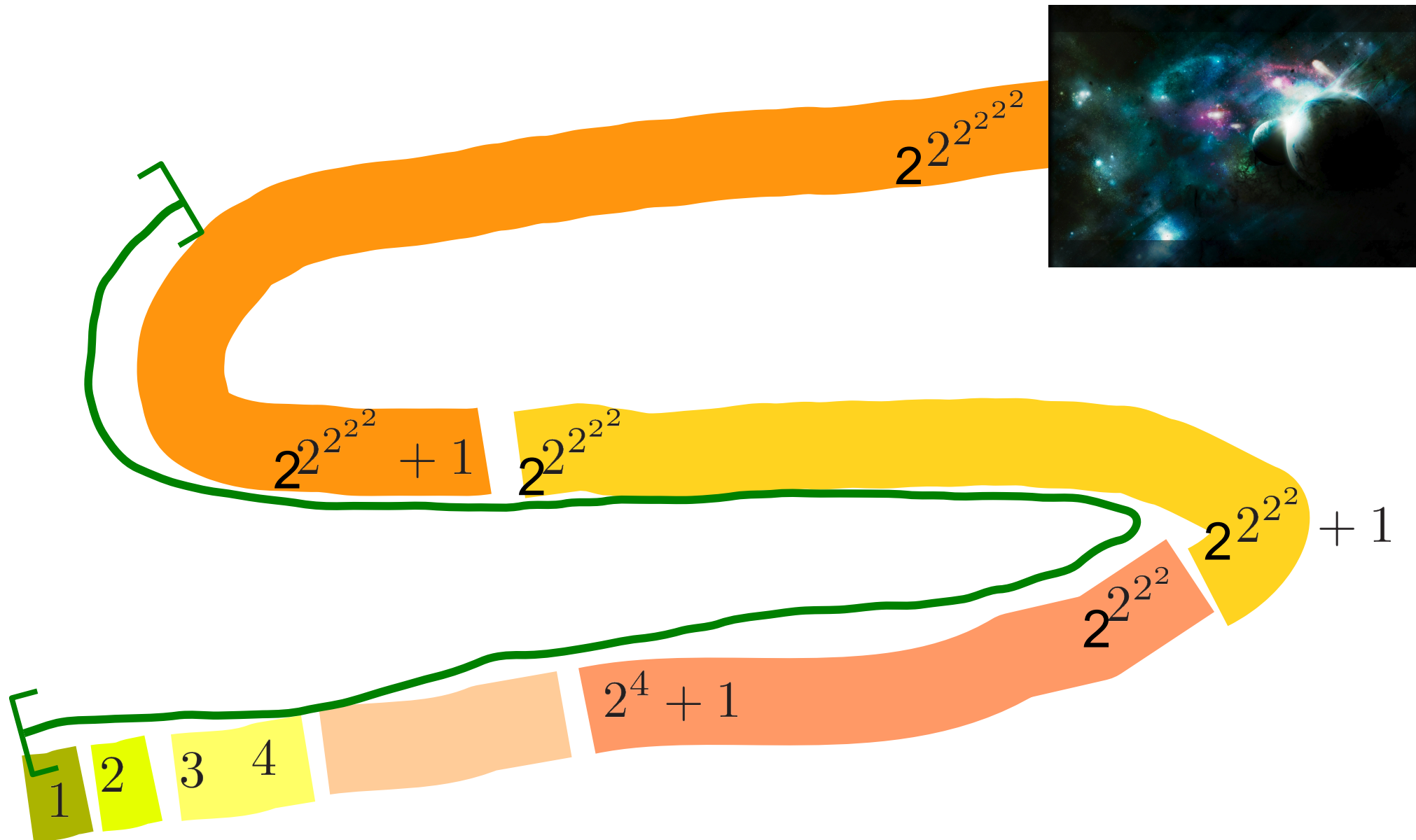
Lors d'une union(x, y)...

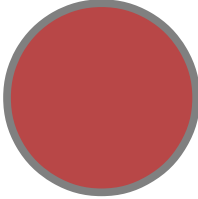


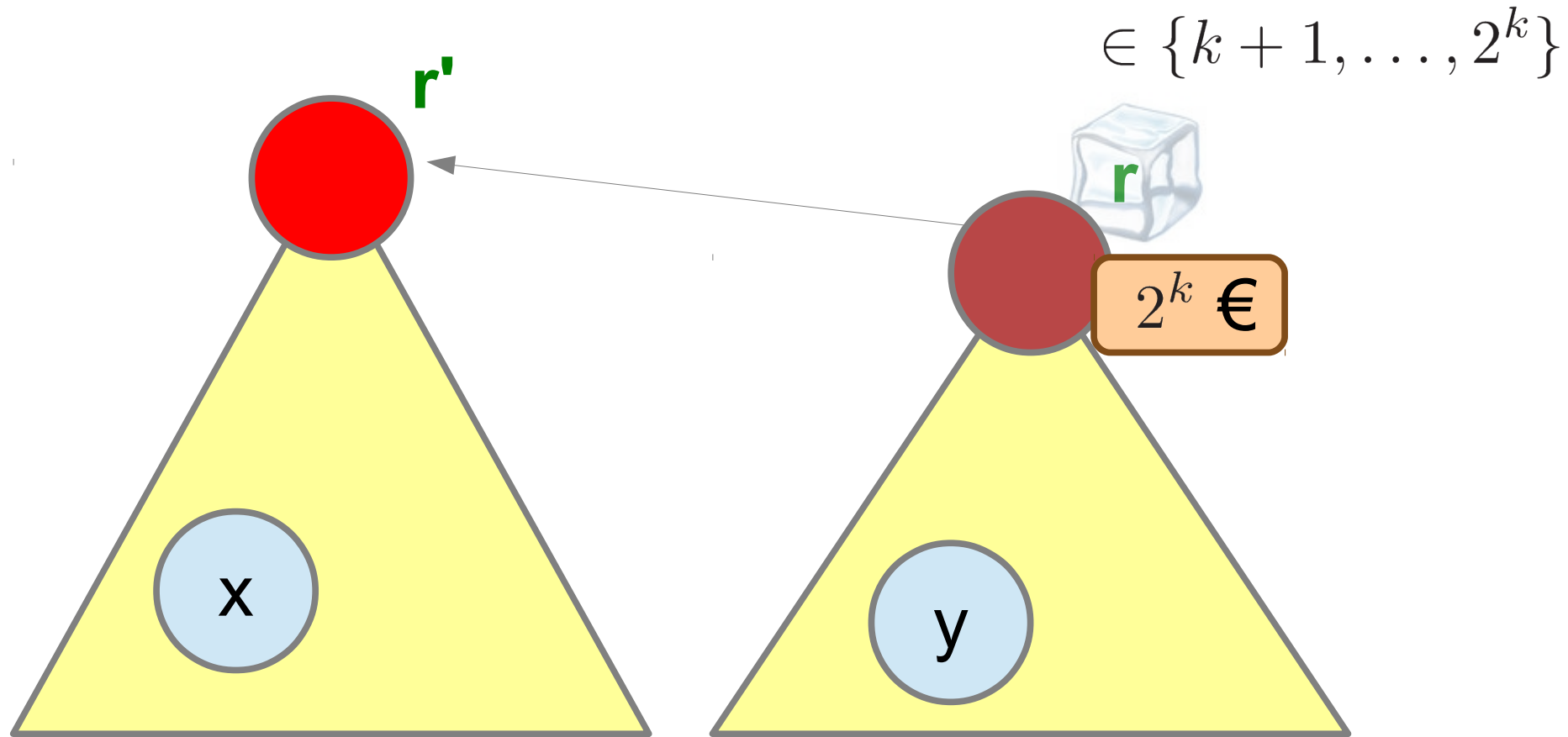
le rang de  ne changera plus.



L'intervalle des rangs possibles découpé en sous-intervalles fous !



le rang de  ne changera plus.

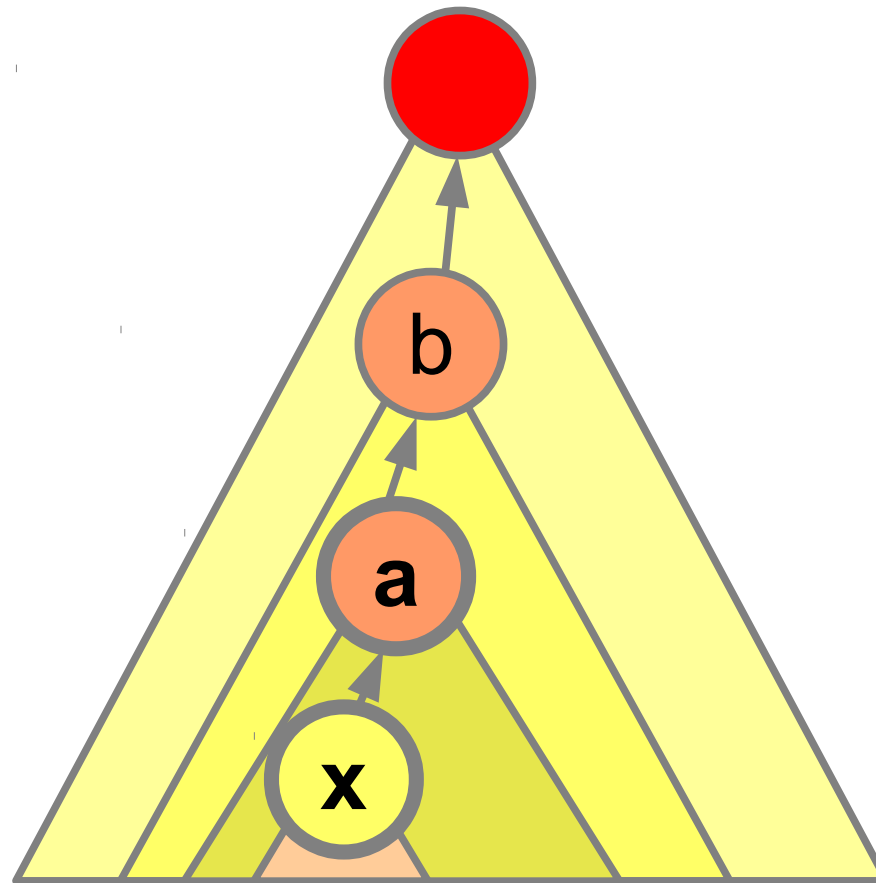
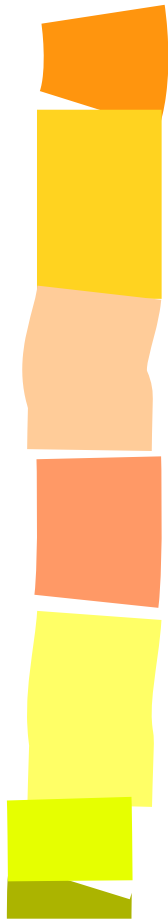


Dettes maximale...

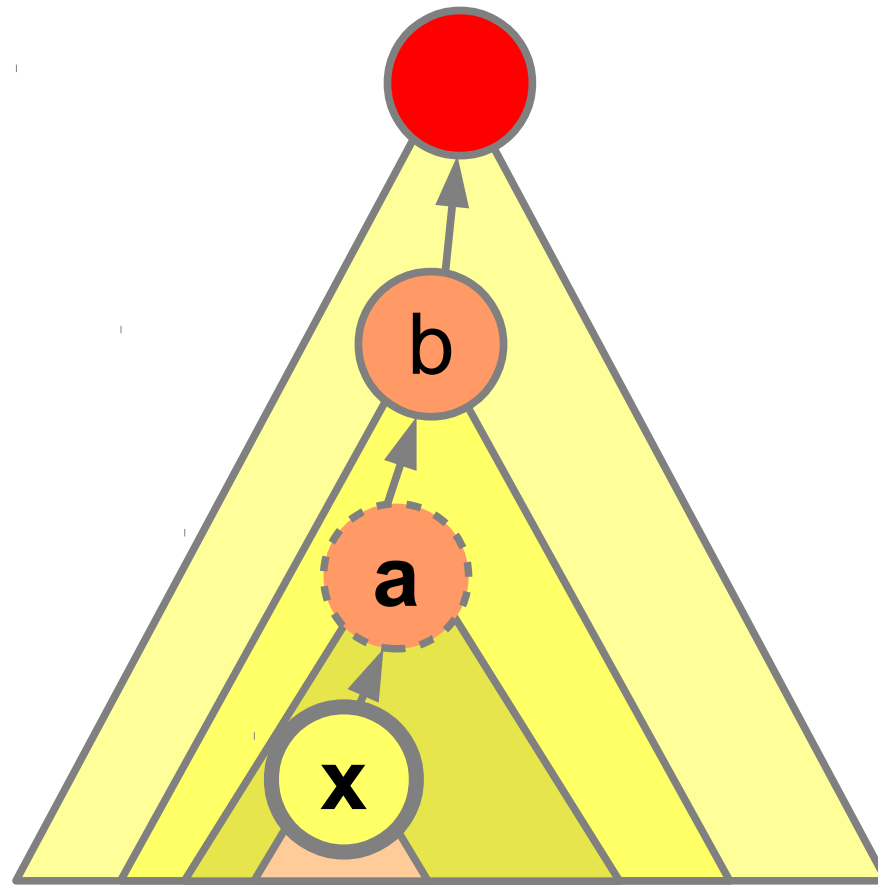
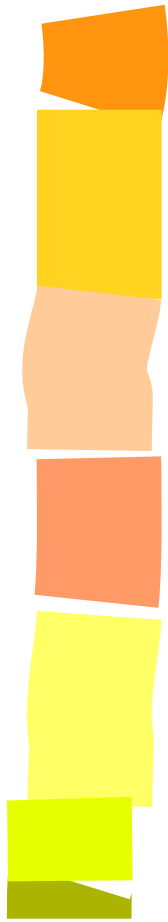
$7n \text{ €}$

au max en pratique...

Find... combien d'opérations ?

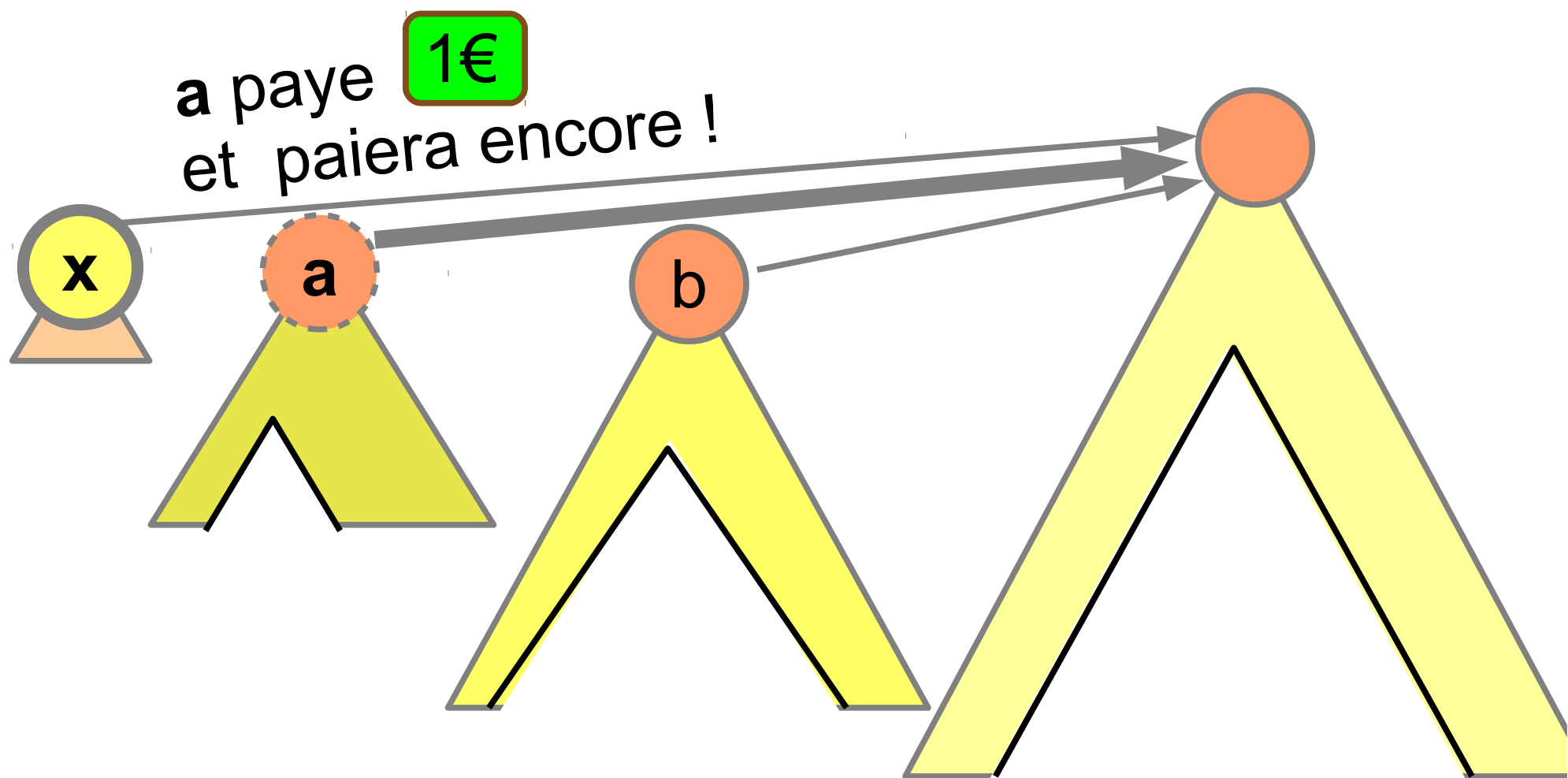


Mais certaines opérations sont comptabilisés comme des paiements !



1€

Va-t-il déboursier tout son argent ?



Personne n'a de dettes !

