**Kruskall**

Le papier parle de l’implémentation de l’algorithme de Kruskal en temps quasi-linéaire.

Kruskall va prendre toutes les arêtes du graphe, dans l’ordre décroissant, et les ajouter au MSP successivement s'ils ne forment pas un cycle.

Pour savoir si une arête peut être insérée dans le MSP, il faut savoir si elle forme un cycle. Un MSP étant une liste, trouver l’appartenance à un ensemble ne se fait pas en O(1). C’est pour cela que l’on va créer une deuxième représentation de ce MSP. On choisit donc de calculer à la fois le MSP, et la structure permettant de tester l’appartenance à un ensemble en temps constant (Union-Find). Ce graphe est nommé QBT.

**Partie 1 : Le calcul en temps quasi-constant**

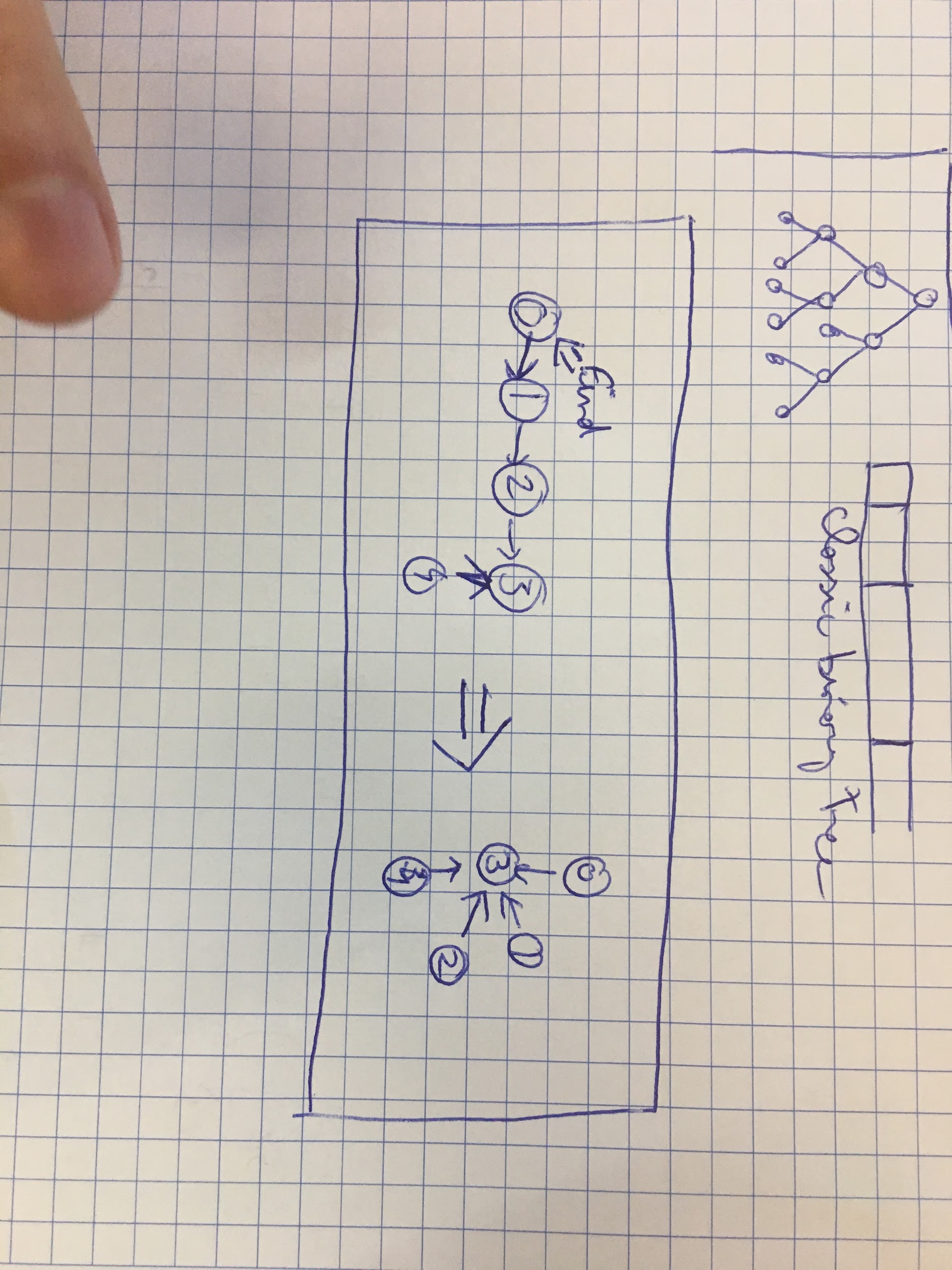
UNION-FIND

Cette structure ne possède qu’une racine par composantes connexe. Cette racine, appelée élément canonique, représente l’ensemble.

2 opérations sont possibles sur les ensembles:

* Find Canonical(c) -> Trouve l’élément canonique de l’élément d’indice c
* Union(cx, cy) -> cy (élément canonique) prends comme père cx

Cette structure est représentée par une array parent[]. Les nodes sont numérotées de 0 à n-1. parent[i] donne l’indice de l'élément parent de la node i.



Voici un exemple d’exécution de la structure union-find. Nous faisons find sur l’élément 0. L’algorithme va “compresser” le chemin allant de 0 à l’élément canonique, permettant de garantir un temps d'exécution quasi-linéaire.

ARBRE

Cette structure est une structure d’arbre binaire de base. Les nodes sont représentées par des indices allant de 0 à n-1. Une array parents[i] définit le parent de la node i.

On a donc deux arrays, qui représentent deux choses différentes. La première n’est en rien un arbre, mais permet de trouver l’appartenance à un ensemble en O(1). Le deuxième représente un arbre binaire => une hiérarchie.

On va donc réunir le meilleur des deux mondes et utiliser EN MÊME TEMPS les deux représentations, pour pouvoir calculer en O(n) un MSP. n étant le nombre d’arêtes.

**Partie 2**

L’autre partie de ce papier donne une méthode pour obtenir la hiérarchie des zones quasi-plates (arbre QCT).

Pour ce faire, on rassemble les sommets de QBT de même poids (sauf racines et feuilles). Cela nous ramène donc à une hiérarchie. On obtient ainsi un nouvel arbre QCT, qui va nous servir par la suite. Cet arbre sera utile au projet.

Ci dessous la transformation effectuée

2 2

/ /

1 1

/ /

1 0

/

0

La dernière partie du papier nous parle d’une hiérarchie d’attributs. Ces attributs correspondent à la surface du graphe QCT. Ils sont calculés récursivement et correspondent à la somme des attributs des enfants. L’attribut d’une feuille correspond à son poids dans QBT ou QCT.