**Summary : « Watershed cuts : Thinnings, shortest path forests, and topological watersheds »**

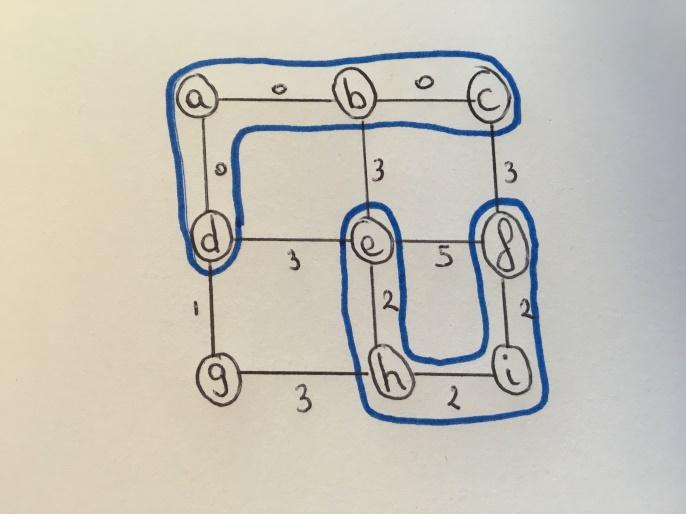
Si l’on fait tomber une goutte d’eau sur le flac d’une montagne, celle-ci ruissellera jusqu’un potentiel « bassin de rétention ». Si l’on fait ruisseler une autre goutte d’eau sur le l’autre flanc de la même montagne, celle-ci rejoindra un autre bassin de rétention. Topographiquement, une ligne de partage des eaux est la ligne séparera les deux flancs. Ainsi si l’on dépose la goutte d’eau d’un côté ou de l’autre de la ligne de partage des eaux, la goutte atteindra un bassin de rétention différent.

Cette notion de ligne de partage des eaux est intéressante dans le domaine du traitement d’image car elle peut permettre de « segmenter » une image, c’est-à-dire de séparer l’image en plusieurs régions logiquement reliées.

Tout d’abord, nous allons travailler avec des « edge weighted graphs », c’est-à-dire que chaque arc possèdera un poids. Mais comment le determiner, puisqu’une image est une suite de pixels, chaque pixel correspondant à un sommet ? Eh bien c’est a notre bon vouloir, en fonction de ce que l’on veut faire sur l’image. Si l’on travaille sur une image en deux dimension (en noir et blanc), un arc peut être simplement la différence entre la valeurs des deux pixels qu’il relie. Si l’on travaille sur une image en couleur, il y aura probablement une autre technique à définir, etc.

Pour déterminer ce qu’est une ligne de partage des eaux algorithmiquement, expliquons les deux notions suivantes :

Soit F un graphe. Un minimum de F est un sous-graphe connexe X de F dans lequel tous les arcs ont un poids égal k (réel positif), et pour lequel tous les arcs adjacents de X sont supérieurs à k.

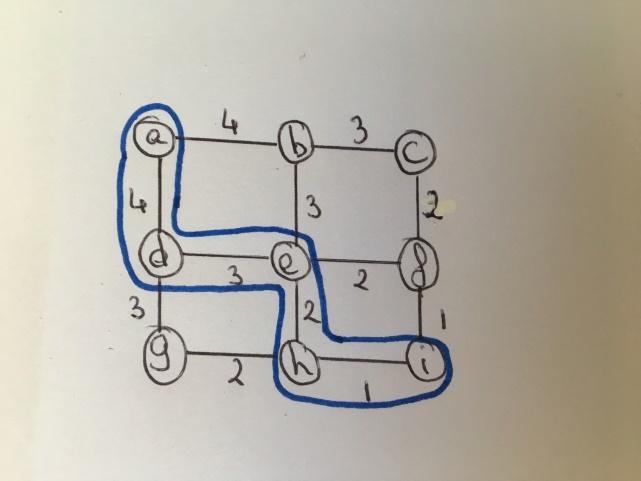
Les minimums d’un graphe F (noté M(F)) est un graphe unissant tous les minimums de F.

*Exemple :*



Un « chemin descendant » est une suite d’arcs reliés entre eux (deux arcs sont reliés entre eux s’ils possèdent un sommet en commun) dont le poids est décroissant.

*Exemple :*



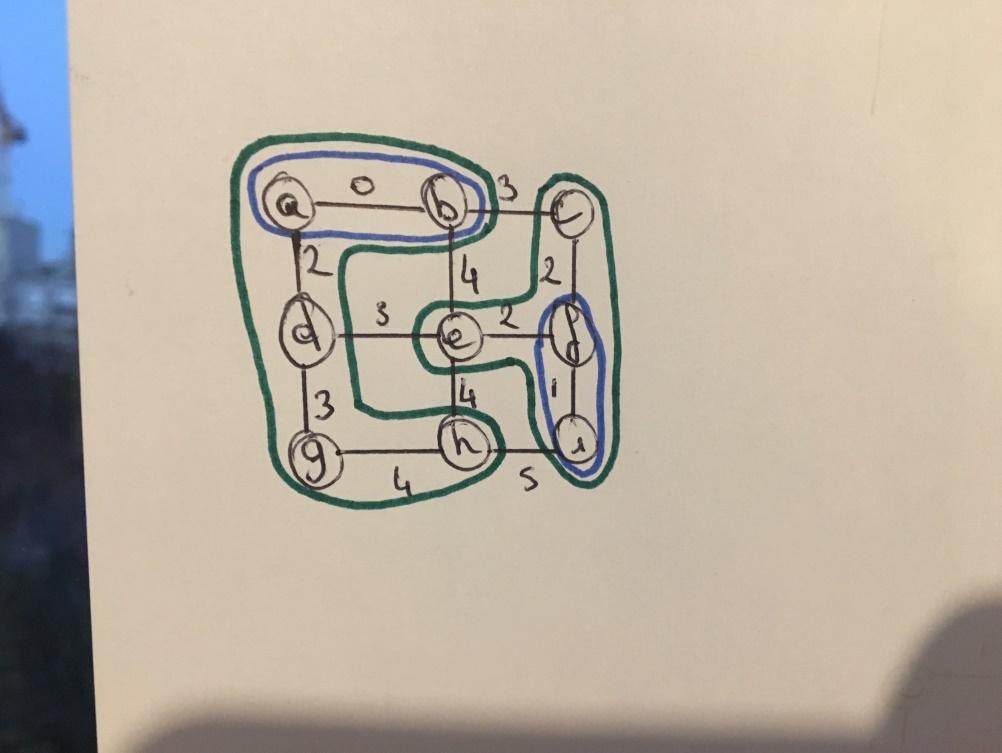
Dans un « edge-weighted graph », (graphe dans lequel les arcs ont un poids (ou une altitude), une ligne des partage des eaux est une suite d’arcs de telle sorte à ce qu’on puisse trouver de part et d’autre deux chemins descendants, menant a deux minimum de F distincts.

Comment peut-on alors déterminer algorithmiquement ces lignes de partage des eaux (ou « watershed cuts ») ?

Notion d’extension d’un graphe : On dit que « Y est une extension de X » si X est inclus dans Y, si Y est connexe, et si tous les sommets et tous les arcs de X sont aussi dans Y.

Si l’on crée des extensions des minimums de F, de telle sorte à ce que chaque sommet de F soient dans une extension d’un des minimums de F un arc reliant deux sommets appartenant à deux extensions différentes est un arc de « coupure ». Intuitivement, il existe au moins une « combinaisons d’extension » de sorte à ce que ces coupures soient des « watershed cuts »

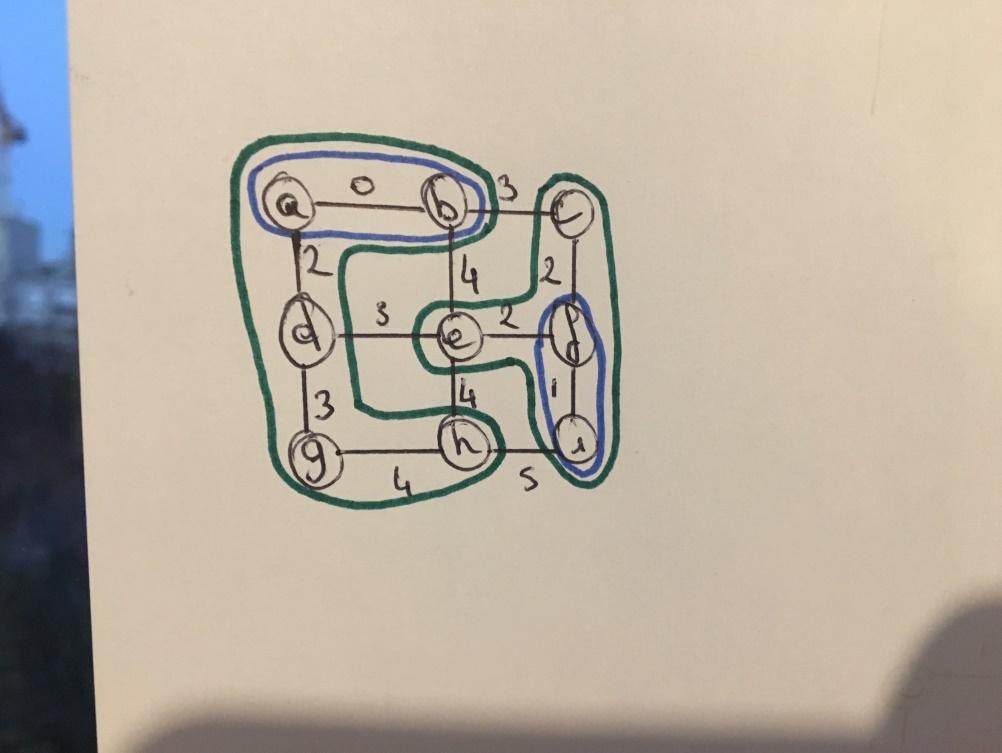
*Exemple :*



Notion de Foret : En théorie, une forêt est un graphe comportant plusieurs arbres (qui ne peuvent pas contenir de cycle). Cependant, comme les minimas de notre graphe F peuvent eux-même contenir des cycles, on dit qu’une forêt est l’union de plusieurs sous-graphes de F connexe dans lequel seuls les minimas de F peuvent contenir des cycles.

Ainsi, une MSF (pour « minimum spanning forest ») relative aux minimums est la forêt dont la somme des poids des arcs la constituant est la plus petite, et dans laquelle chaque arbre est une extension d’un minima.

Dans l’article, on démontre que si l’on trouve la coupure d’un MSF, celle-ci est elle-même une ligne de partage des eaux. Trouver une ligne de partage des eaux revient donc à trouver une MSF.



Pour trouver cette MSF, il y a trois méthodes. Chaque méthode sera utilisée pour un cas différent. La première peut être utilisée dans le cas de calculs « en parallèle », la seconde ne peut être exécutée en parallèle, mais est en complexité linéaire (O(n)), et la dernière utilise le principe d’immersion.

Avant d’expliquer chaque méthode, quelques notions sont à expliquer.

Tout d’abord, nous attribuons à chaque sommet de F la valeur du plus petit arc adjacent à ce sommet. On note F-(x) la valeur ainsi attribuée au sommet x.

Notion de « lowering d’arc ». Le lowering d’un arc u(x,y) correspond au remplacement du poids de u par le minimum de F-(x) et de F-(y). En d’autres termes u prendra le poids de l’arc adjacent ayant le poids minimal.

Intuitivement, il est compréhensible qu’on ne puisse pas faire de « lowering » de tous les arcs, ça ne mènerait à rien. Ainsi, pour sélectionner les arcs à « abaisser » nous allons vérifier une propriété sur chaque arc. Si cette propriété retourne « True » sur l’arc u, l’abaissement sera effectué sur u, dans le cas contraire, il ne sera pas fait.

A – Première propriété : Border-thinning

Grace à l’ajout des poids sur les sommets, on peut catégoriser les arcs en 3 sortes.

Les arcs « séparant localement » sont les arcs dont le poids est supérieur aux deux sommets qu’ils relient.

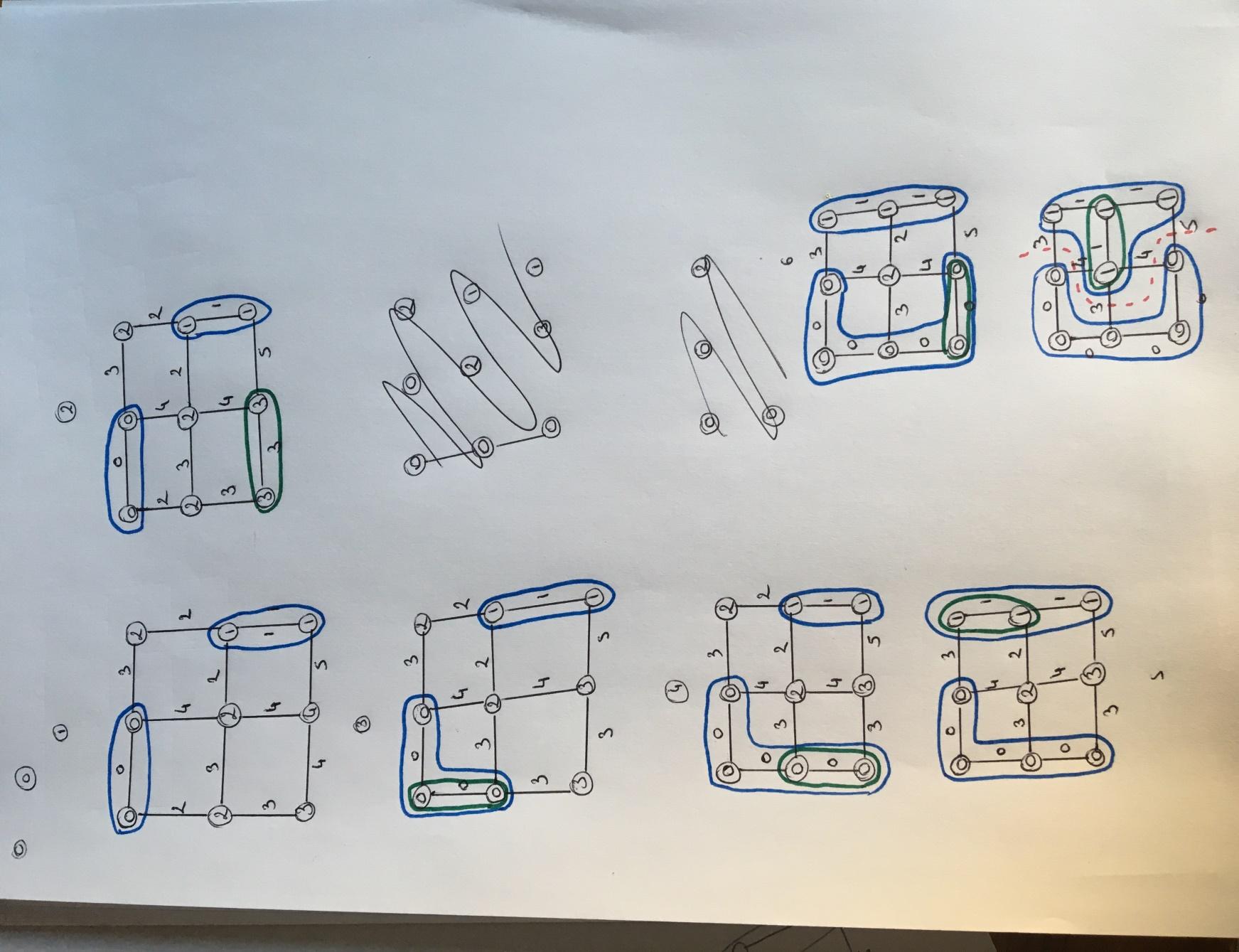
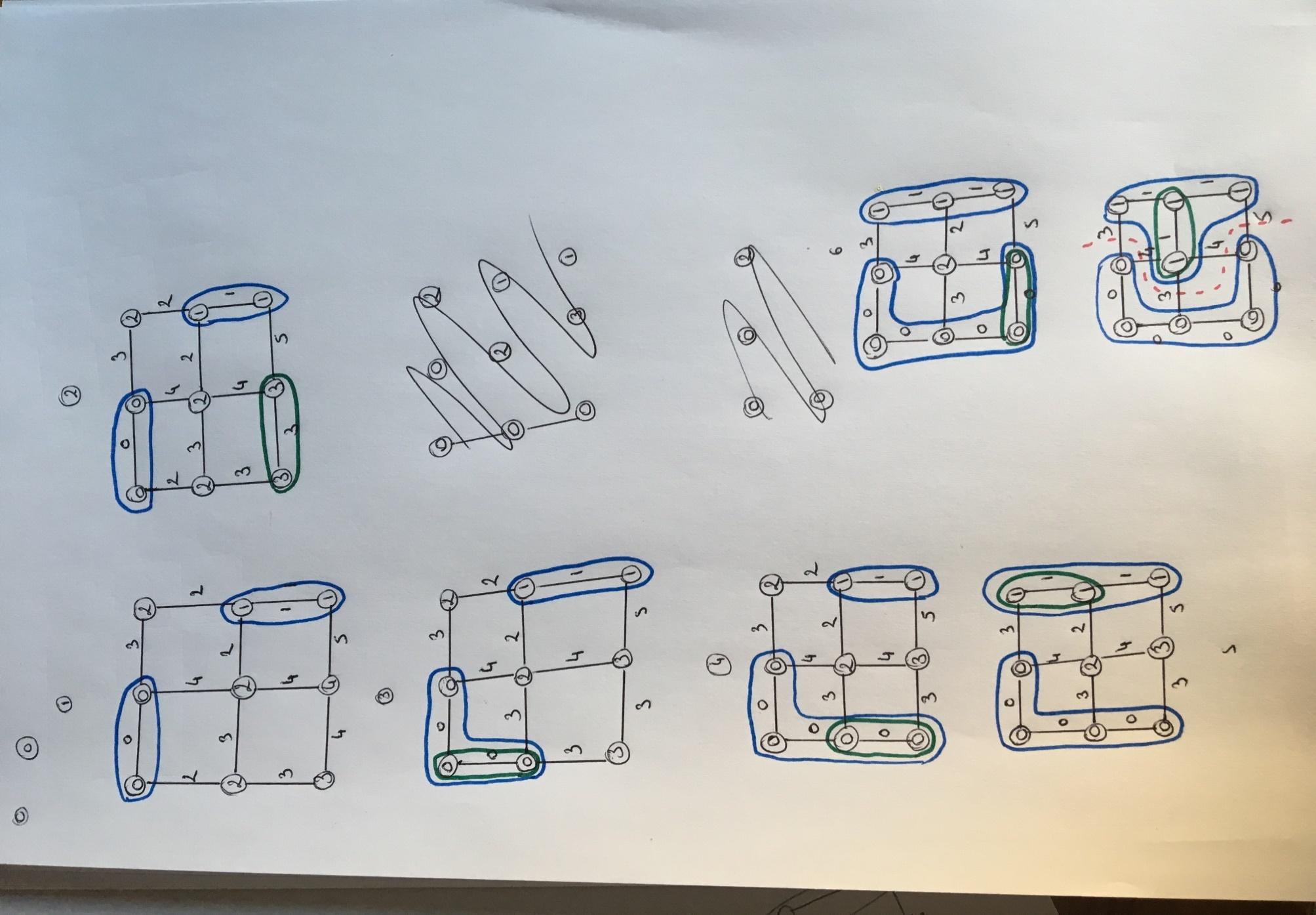
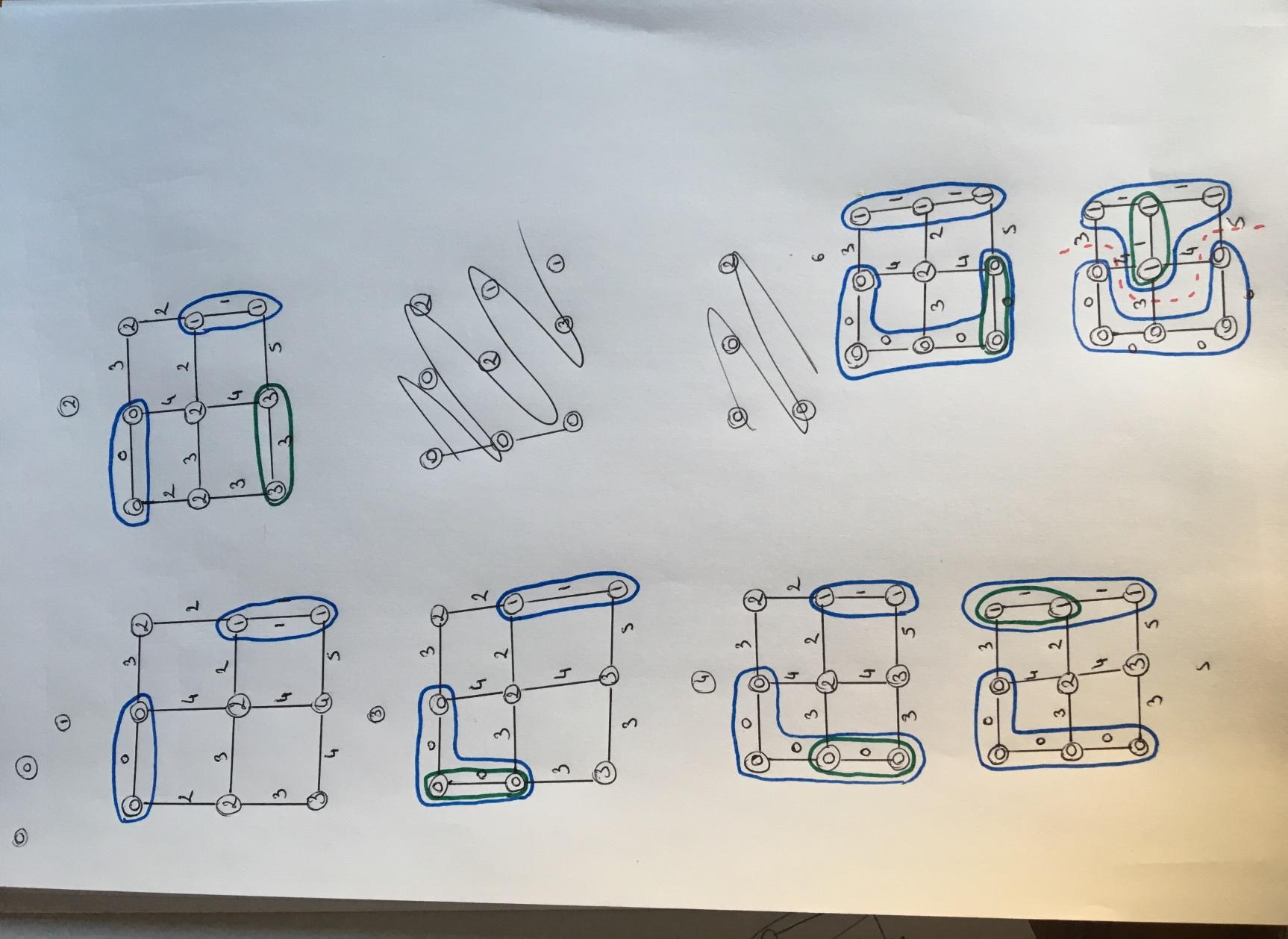
Les arcs « internes » sont égaux aux deux sommets qu’ils relient. C’est par exemple le cas des arcs à l’intérieur des minimums de F

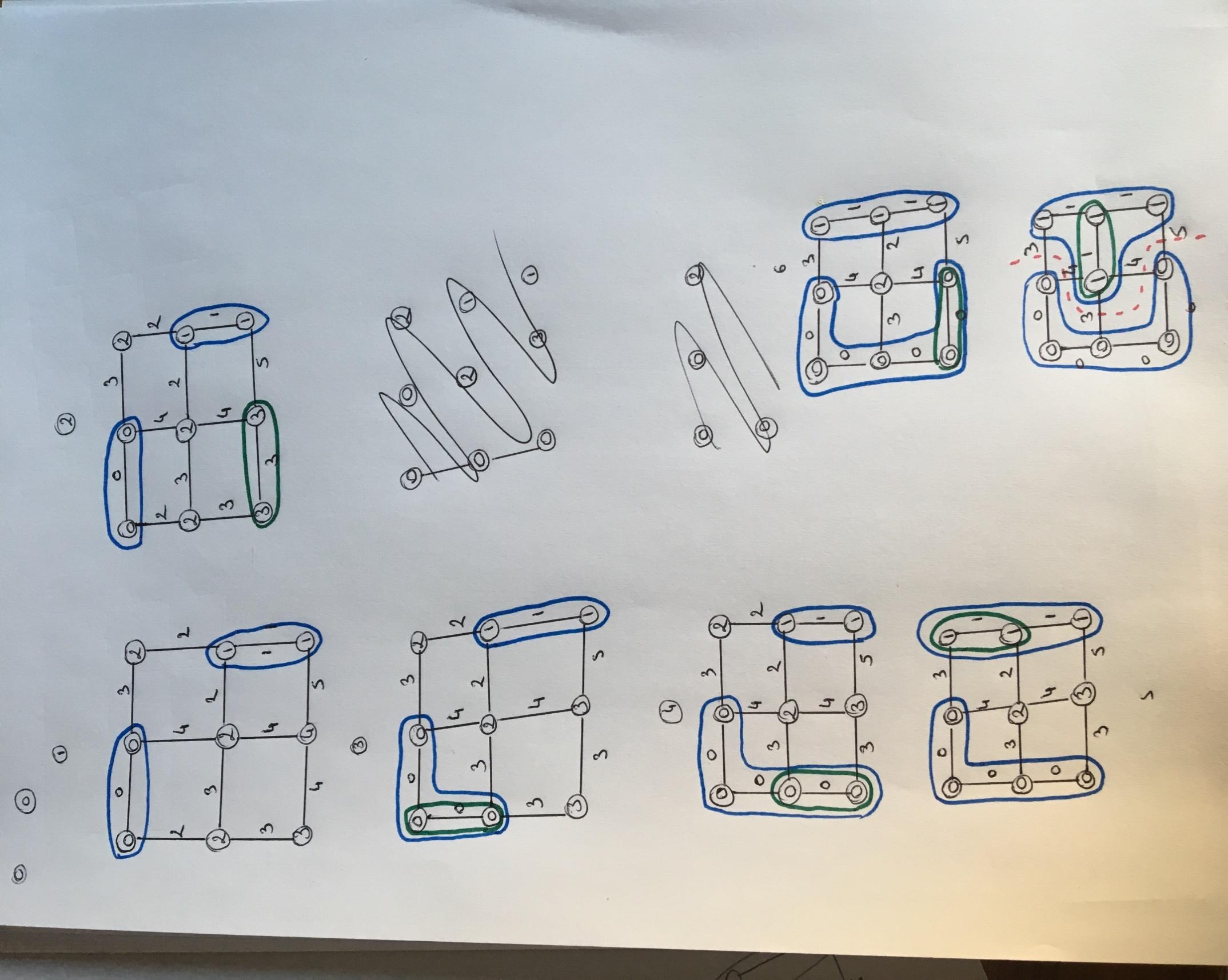
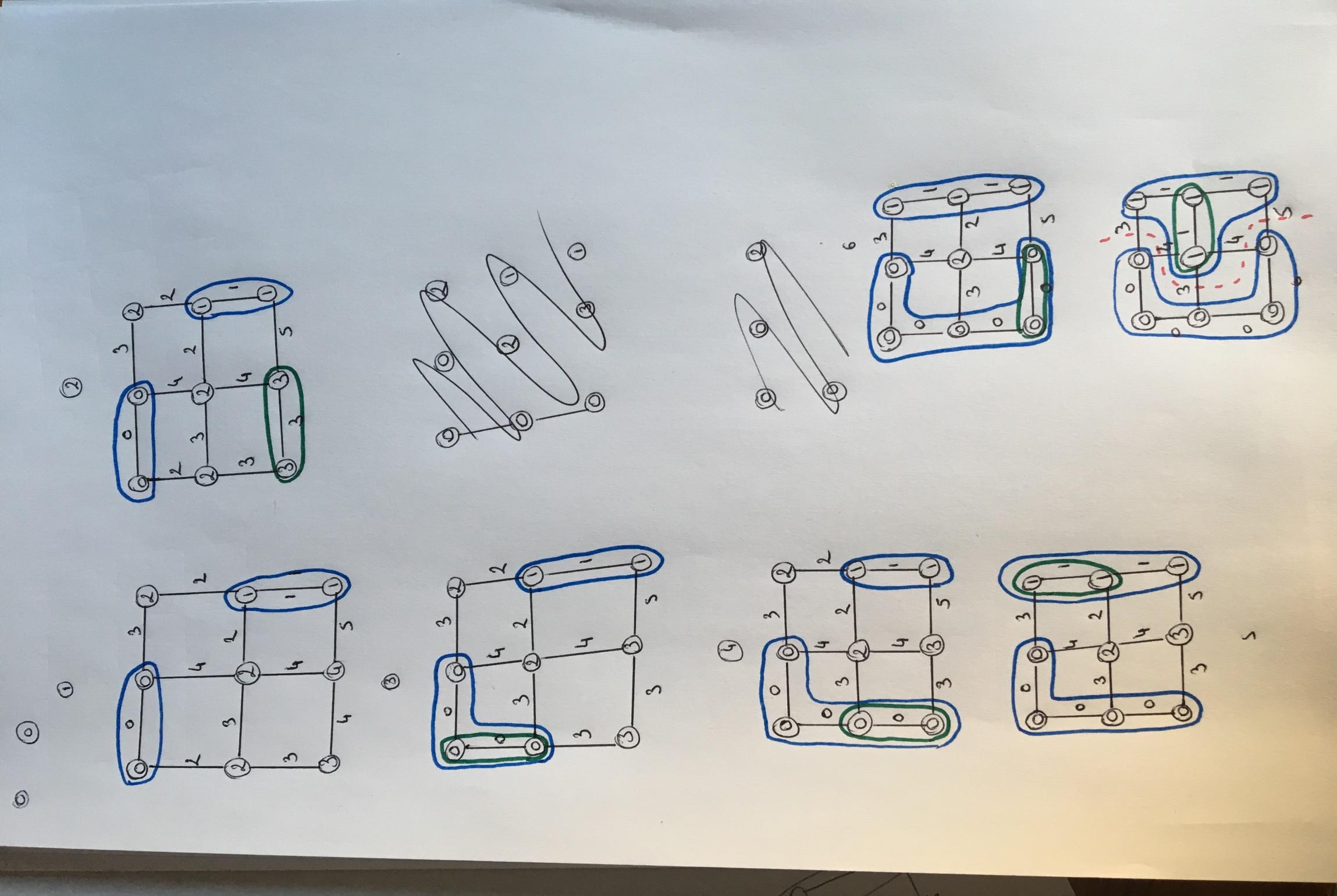
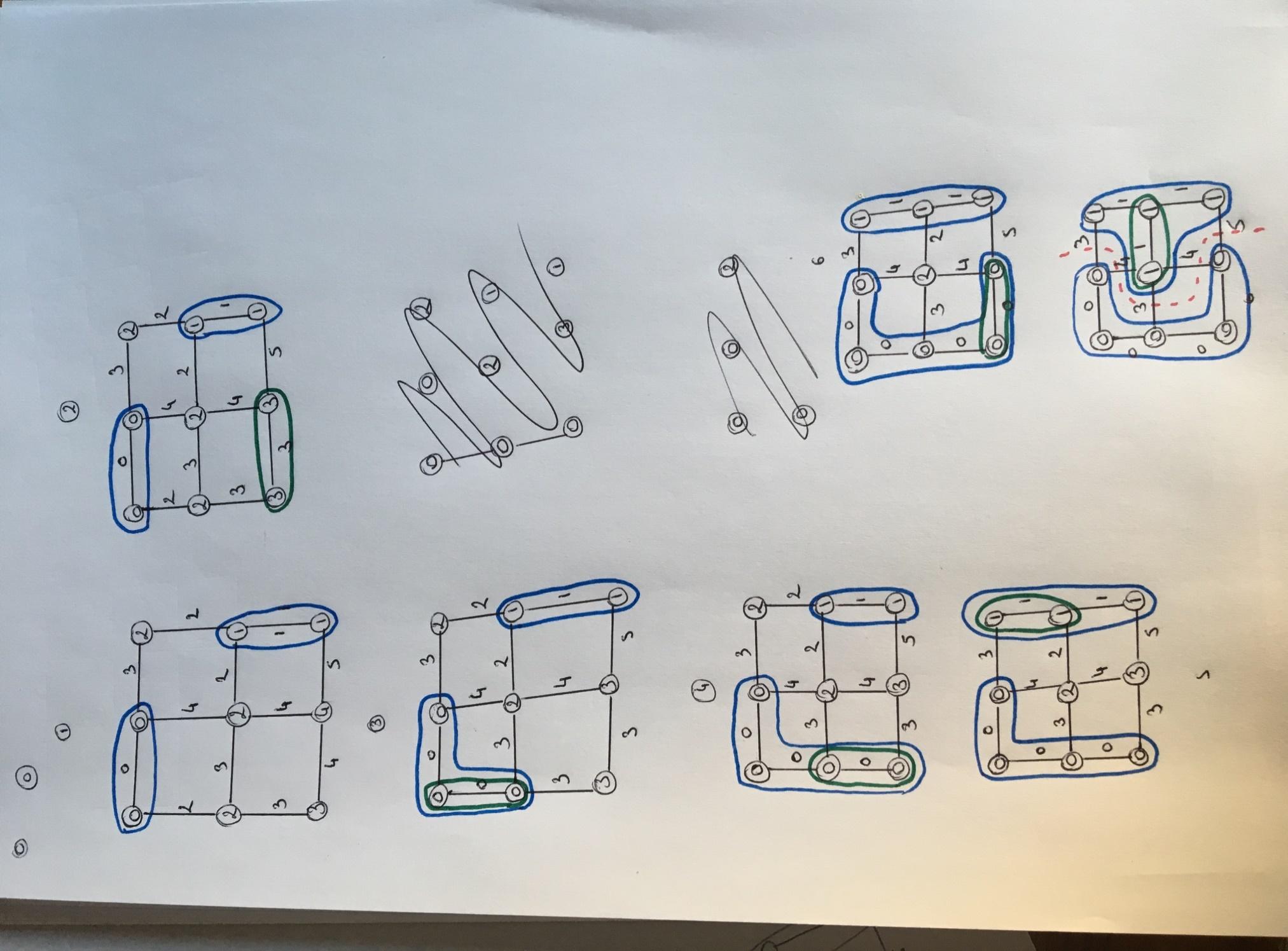
Les arcs de bordures sont les arcs qui sont égaux à l’un des sommets qu’ils relient, et supérieur à l’autre sommet.

Puisque le poids des sommets est le minimum des arcs adjacents à ce sommet, un arc ne peut pas être inférieur au poids d’un sommet adjacent. Les 3 catégories ci-dessus couvrent donc bien l’entièreté des possibilités.

La propriété de lowering est donc la suivante. Un arc devra être « abaissé » si et seulement si il appartient à la catégorie « arc de bordure ». Tant qu’il existe au moins un arc qui respecte cette propriété (c’est-à-dire, tant qu’il existe un arc appartenant à la catégorie « arc de bordure ») , le processus de lowering devra être poursuivi. Ce processus est appelé le *B-Thinning* (pour *Border-Thinning*) et lorsqu’il sera terminée, graphe obtenu, qui ne pourra plus être abaissé d’avantage, est un graphe dont l’union des minimum est appelé *B-Kernel*. (l’abaissement successif aura abaissé les poids d’un grand nombre d’arc de telle sorte à ce que tous les sommets fassent partie d’un B-Kernel). Le B-Kernel est tout simplement le MSF de F.

*Exemple en étapes : pour chaque étape : en bleu les minimas, en vert, l’arc de bordure « abaissé »*





Cette méthode est intéressante du fait qu’elle est executée en local (sur chaque arc, sans se soucier du reste). Elle peut ainsi être exécutée en parallèle.

Cependant, si l’execution en parallèle n’est pas désirée, on se rend compte que la complexité de l’algorithme est est bien trop élevée (O(E2)). E (pour « edge ») étant le nombre d’arc du graphe.

B – Deuxième propriété : Minimum-thinning

Cette propriété est un cas particulier de B-Thinning. Si l’on exécute « à la main » un B-Thinning, on se rend compte très rapidement que si on vérifie la propriété (d’arc de bordure) seulement sur les arcs adjacents aux minimums de F, le lowering ne sera effectué qu’au plus 1 fois sur chaque arc…

Par exemple, sur l’exemple précèdent (B-Thinning), on comprend rapidement que l’étape deux est de trop, qu’elle ne sert pas réellement a quelque chose, puisque l’arc est a nouveau modifié à l’étape 5…

Ainsi, l’efficacité du *B-thinning* est augmentée est l’algorithme ne parcourra qu’une seule fois tous les arcs de F. On obtiendra donc un algorithme en complexité linéaire, pour des résultats similaires au *B-Thinning*

C – Troisième propriété : Immersion-thinning

Cette méthode est la méthode la plus utilisée. Si l’on devait faire une image topographique, cette méthode consiste à inonder le graphe, en mettant des barrage de telle sorte a ce que l’eau de deux bassin de rétention différents ne soit pas mélangée.

Algorithmiquement, cette méthode tient en trois étapes :

1. Marquer les sommets des minimums de F avec des labels différents.
2. Prendre l’arc contenant 1 et 1 seul sommet ayant un label (s’il y a plusieurs arcs dans ce cas, prendre l’un d’entre eux peu importe lequel), puis « abaisser » cet arc.
3. Répéter l’étape 2) jusqu’a ce que tous les sommets aient un label.

A chaque étape, le graphe vérifie la propriété *I-Thinning*, et lorsque l’algorithme est terminé, on obtient un *I-Kernel* correspondant à la MSF de F.