

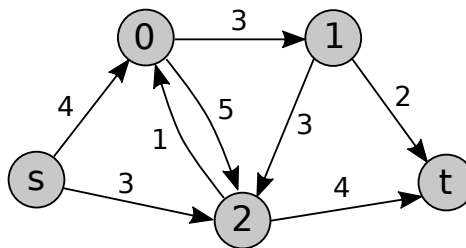
Vision par ordinateur

TD 2 - Segmentation binaire par graphcut

1 Prise en main du projet

Sous CodeBlocks, ouvrez le projet **graphcut.cbp**. Les dossiers `opencv-3.4.1-x64` et `qtbase-small-5.6.3-x64` doivent être à côté du dossier contenant le projet (si ce n'est pas le cas, vous devrez modifier les chemins pour le compilateur et le linker dans les *Build options* du projet). CodeBlocks doit être configuré de manière à utiliser le compilateur MinGW-w64 (comme indiqué dans l'énoncé du TP de traitement d'images avec OpenCV).

La classe `GraphFlow` permet de construire un graphe de flot, comportant au moins un nœud source et un nœud puits (*sink*). Chaque arc est doté d'une capacité. Par exemple, le graphe de flot ci-dessous,



peut être construit de la façon suivante :

```
GraphFlow G;
G.setNbNodes(3);
G.connectNodes(0, 1, 3); G.connectNodes(0, 2, 5);
G.connectNodes(2, 0, 1); G.connectNodes(1, 2, 3);
G.connectSourceToNode(0, 4); G.connectSourceToNode(2, 3);
G.connectNodeToSink(1, 2); G.connectNodeToSink(2, 4);
```

La méthode de Ford-Fulkerson, qui détermine à chaque itération un chemin augmentant, est partiellement écrite. Le calcul de la coupe s/t à partir du flot maximal est écrite en totalité.

2 Exercice

Votre travail devra être rendu à la fin de la séance par mail. Vous mettrez directement en pièce jointe les fichiers **graph.cpp** et **main.cpp**, préalablement renommé en **graph_VOTRENOM.cpp** et **main_VOTRENOM.cpp**. Vous ne mettrez d'autres fichiers **.h/.cpp** que si vous les avez modifiés.

Le projet contient 3 TODO. Vous allez devoir compléter le code :

- TODO 1 : construction du graphe de flot pour la segmentation binaire
- TODO 2 : dans l'algorithme de Ford-Fulkerson, calculer la capacité résiduelle du chemin augmentant et mettre à jour le flot sur les arcs de ce chemin
- TODO 3 : remplir l'image de segmentation (*labels*) à partir de l'ensemble S de la coupe s/t