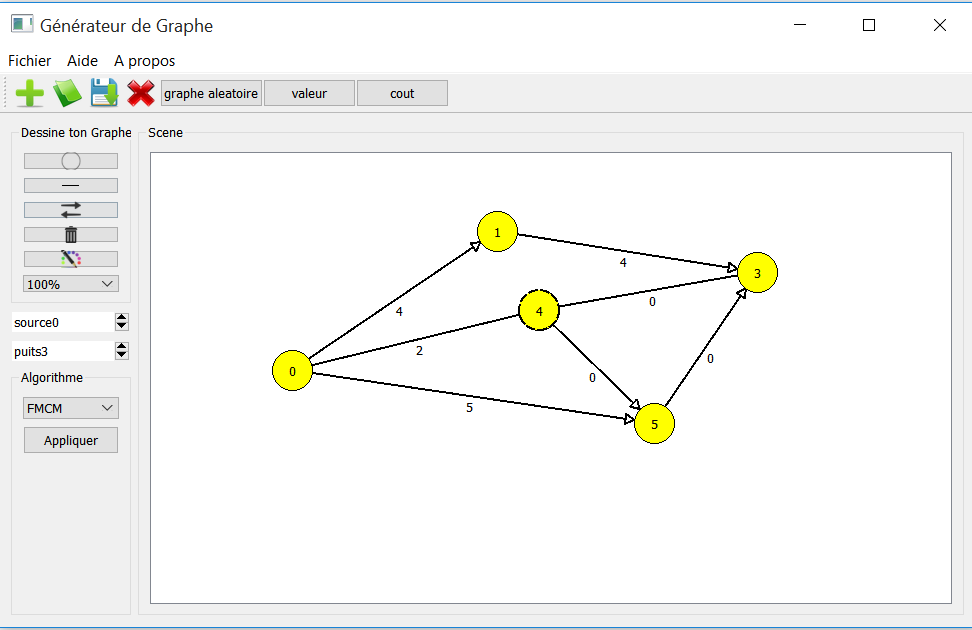


**Rapport final :**



Mme ZRIKEM Maria

Mr EL MARZOUKI Nabil

Mme BENCHIKHI Loubna

Les encadrants

Les intervenants

MBARKI Anis

FILAH Anas

KADIM Mouad

IBNOUKHALKANE Tarik

OUARHOU Lhoussaine

SOMMAIRE

[1-Contexte gLOBALE du projet 3](#_Toc517270108)

[1-1. Introduction 3](#_Toc517270109)

[1-2. Objectifs 4](#_Toc517270110)

[1-3. Planning 5](#_Toc517270111)

[2-Étude technique : 7](#_Toc517270112)

[3-Interface : 9](#_Toc517270113)

[Les Elements de l’interface: 9](#_Toc517270114)

[MenuBAR 10](#_Toc517270115)

[ToolBAR 11](#_Toc517270116)

[Groupbox 13](#_Toc517270117)

[scene 15](#_Toc517270118)

[le fonctionement du programme 15](#_Toc517270119)

[4-structure de projet 30](#_Toc517270120)

[5-Les methodes utilisees dans chaque classe : 31](#_Toc517270121)

[6-Generalité et étude fonctionnelle 34](#_Toc517270122)

[DESCRIPTION du Programme: 34](#_Toc517270123)

[7-Initialisation du problème : 35](#_Toc517270124)

[8- problème de flot : 36](#_Toc517270125)

[Théorème de Ford-Fulkerson 36](#_Toc517270126)

[Flot max 37](#_Toc517270127)

[flot max a cout minimum 40](#_Toc517270128)

[utilisation de l’algorithme de ford 41](#_Toc517270129)

[9-application des algorithmes : 42](#_Toc517270130)

# 1-Contexte gLOBALE du projet

## Introduction

Dans le cadre du projet du second semestre de la 1ere année, nous devons réaliser un Générateur de graphe avec le problème de flot comme sujet. Le but de ce projet et de tester les connaissances de l’étudiant ainsi que de mobiliser ses compétences acquises jusqu'à présent.

Ce rapport est le résultat d’un travail collectif.

On commencera par présenter l’interface, le générateur et les algos.

## 1-2. cahier de charge

* Générateur de graphe : de toutes tailles, tenir compte de la densité ……
* Un outil graphique de traçage et visualisation de graphe
* Opérations de base sur les graphes : ajout et suppression de sommets, d’arêtes ….
* Le graphe peut être :
  + saisi directement par l’outil graphique ensuite stocké dans la structure de données,
  + produit par le générateur
  + lu à partir d’un fichier et visualiser par l’outil graphique
* Flot max et flot max à cout minimum et floit max sous contraintes de capacité.

## 1-3. Planning

Etape 1 : Réalisation de l’interface sans utiliser Qt designer.

Etape 2 : Réaliser le générateur manuel et automatique, avec la possibilité de supprimer les sommets, arcs ou arêtes, et d’enregistrer le projet.

Etape 3 : Implémenter les Algorithmes de flot.

*Planning de la realisation du projet sur les graphes*

* **Le mercredi 18 Avril** :

1) présenter à quoi va ressembler l’interface.

2) Collecter des infos sur les options que présente l’application

* **Le mercredi 25 Avril :**

Présentation du planning

* **le mercredi 2 mai :**

1) Présenter une application qui permet de dessiner des arcs, des arêtes

Et des sommets, ainsi que de créer des nouvelles pages vides.

2) Proposer une manière de sauvegarde, sur papier.

* **Le mercredi 9 mai :**

1. Présenter un générateur qui a toute les fonctionnalités

Fondamentales (créer des nouvelles pages, y dessiner des graphes

Quelque soit la figure voulu, et pouvoir les sauvegarder pour une

Utilisation future).

* **Le mercredi 16 mai :**

1. Ajouter des options à l’application et finaliser le générateur,

la possibilité d’affecter des valeurs au arcs et au arêtes.

2) Présenter des études et des recherche sur le flot, afin d’établir

L’adéquat algorithme pour gérer les flots.

* **Le mercredi 30 mai :**

1) Traduire l’algorithme au langage C++(Qt).

2) Avoir une application apte de resoudre le problème des flots

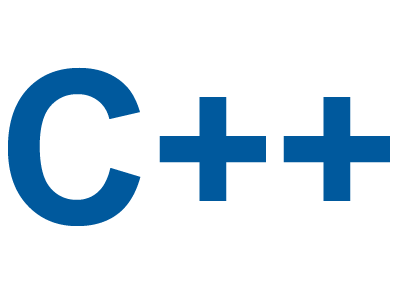
* **Le mercredi 06 juin :**

1. Présenter une application qui peut générer les graphe et traiter les

Flots.

# 2-Étude technique :

Lors de notre projet, nous allons utiliser :



* Les bibliothèques de C++.
* Le développement va être effectué avec QtCreator.

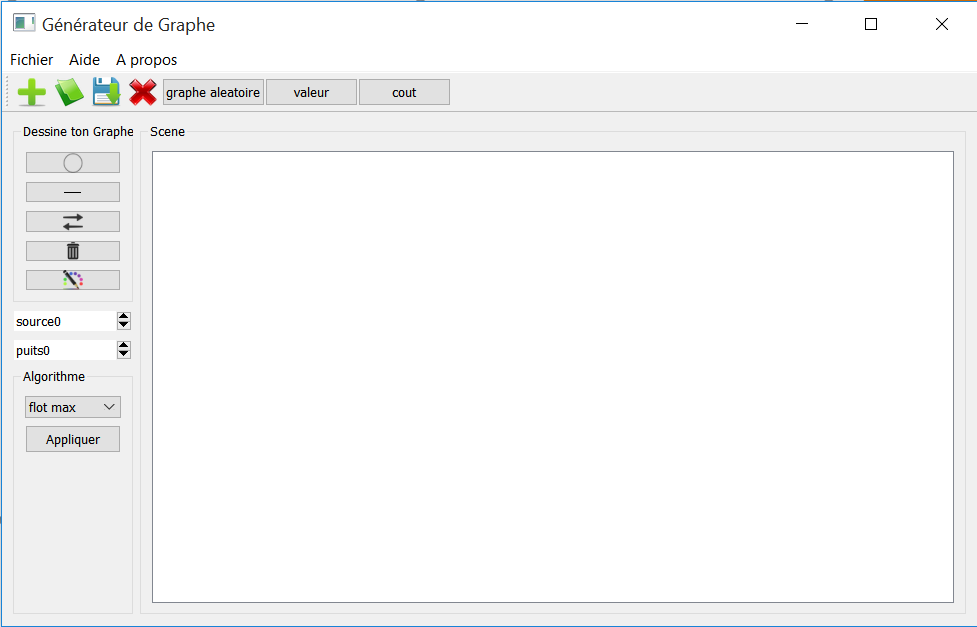
Qt est écrite en C++ et elle est, à la base, conçue pour être utilisée en C++. Toutefois, il est aujourd'hui possible de l'utiliser avec d'autres langages comme Java, Python, etc.

Plus fort qu'une bibliothèque : un Framework

Qt (voir logo en figure suivante) est en fait… bien plus qu'une bibliothèque. C'est un ensemble de bibliothèques. Le tout est tellement énorme qu'on parle d'ailleurs plutôt de **framework** : cela signifie que vous avez à votre disposition un ensemble d'outils pour développer vos programmes plus efficacement

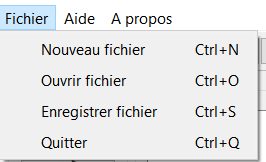
**C++** est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [compilé](https://fr.wikipedia.org/wiki/Compilateur) permettant la programmation sous de multiples [paradigmes](https://fr.wikipedia.org/wiki/Paradigme_(programmation)) (comme la [programmation procédurale](https://fr.wikipedia.org/wiki/Programmation_proc%C3%A9durale), [orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) ou [générique](https://fr.wikipedia.org/wiki/Programmation_g%C3%A9n%C3%A9rique)). Ses bonnes performances, et sa compatibilité avec le [C](https://fr.wikipedia.org/wiki/C_(langage)) en font un des langages de programmation les plus utilisés dans les applications où la performance est critique.

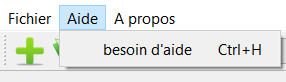
# 3-Interface :

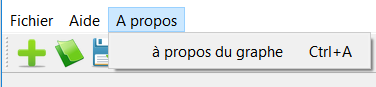


## Les Elements de l’interface:

### MenuBAR







**Menu fichier:** Permet d’afficher des fonctionnalités qui permettent de créer un fichier, d’ouvrir un fichier, d’enregistrer un fichier et de quitter le programme.

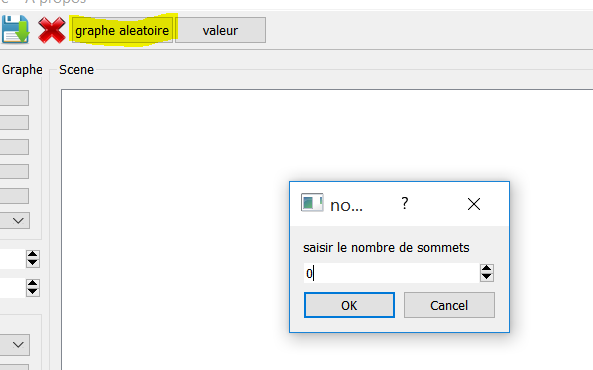
Menu Aide: Permet de donner quelques informations afin d’utiliser les fonctionnalités du programme.

Menu A propos : Permet d’afficher quelques informations sur le projet.

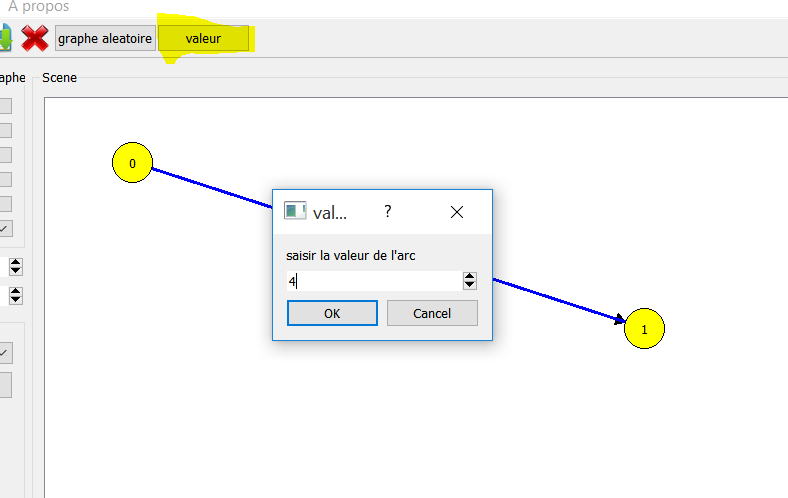
### ToolBAR

C:\Users\ASUS\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6.png

Le Toolbar contient 4 boutons qui permettent d’ouvrir un nouveau fichier, d’ouvrir un fichier, de sauvegarder un fichier ou de quitter le programme.

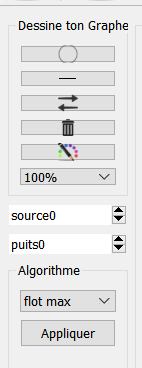


Le Toolbar contient aussi un widget « graphe aleatoire »qui permet d’afficher 3 QInputDialog qui permettent d’entrer le nombre de sommets, d’arêtes et d’arcs. Lorsqu’on appuie sur le bouton OK, le graphe sera générer d’une façon aléatoire.

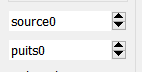


Le Toolbar contient aussi un widget « valeur » qui permet d’afficher un QInputDialog afin de donner la capacité d’un arc ou d’un arête.

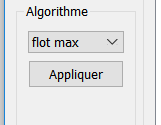
### Groupbox



Le groupbox contient des boutons qui permettent de dessiner le sommet, l’arête et l’arc, de supprimer des éléments de la scène ou de les sélectionner.

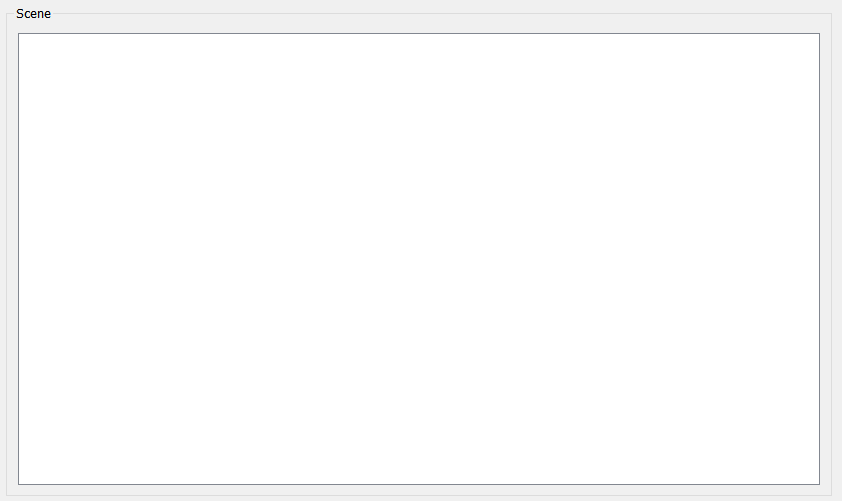


Les deux Qspinbox permettent de designer le sommet qui sera considérer comme source, et celui qui sera considérer comme puits.



Le flot box permet de choisir l’algorithme qu’on veut utiliser, et on appuie sur Appliquer pour l’appliquer sur le graphe déjà réaliser.

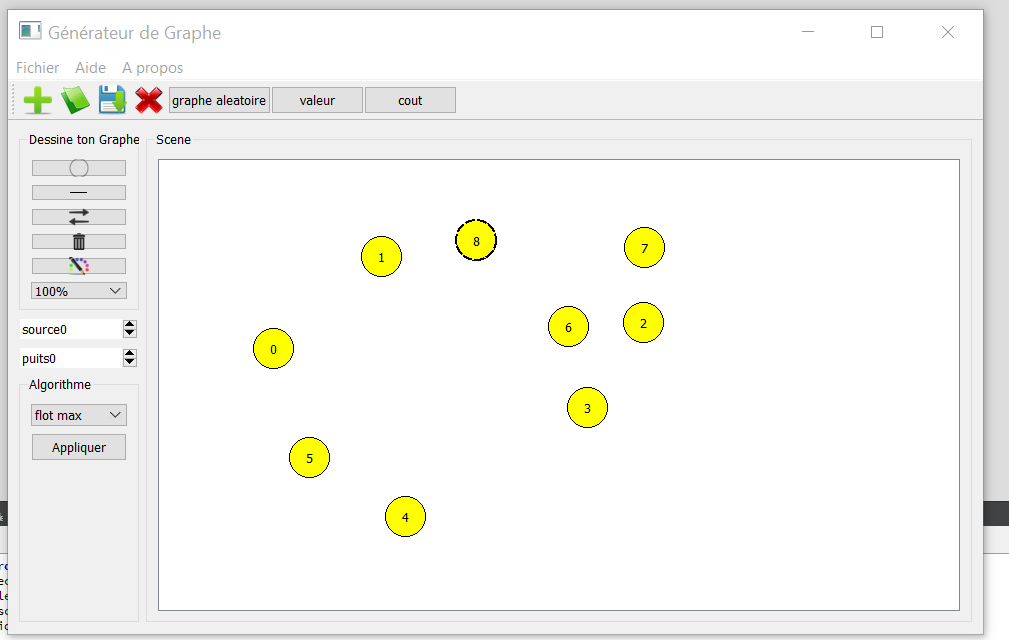
### scene



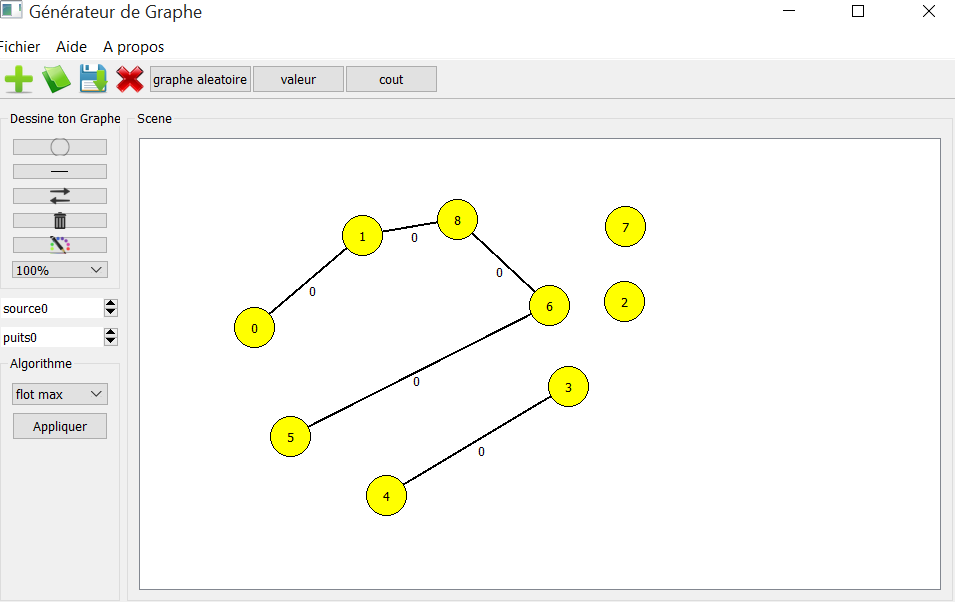
La scène est l’emplacement où l’utilisateur peut dessiner son graphe

## le fonctionement du programme

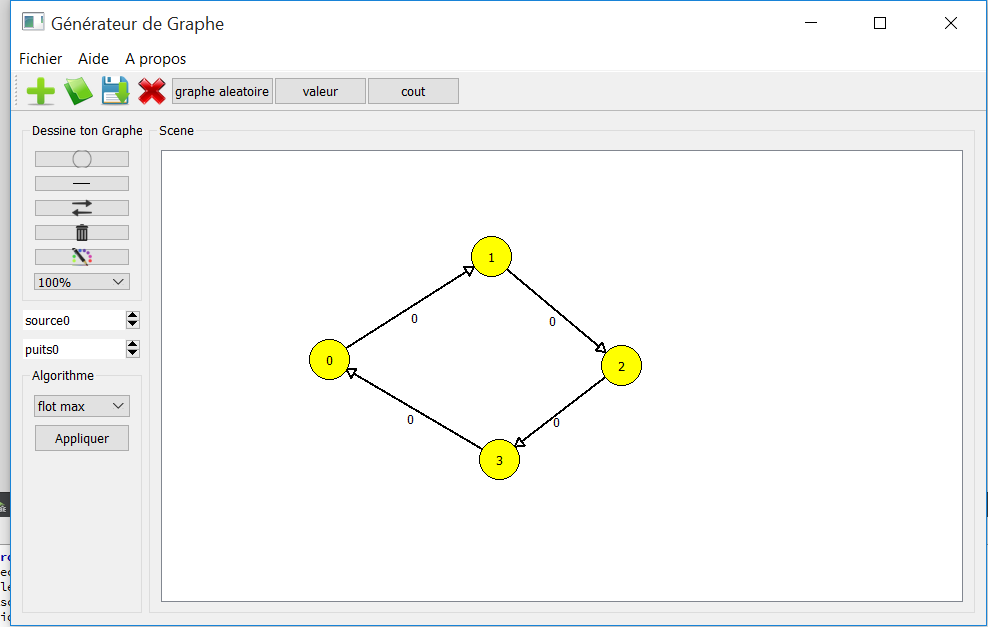
###### ajout d‘un sommet



###### ajout d’une arete



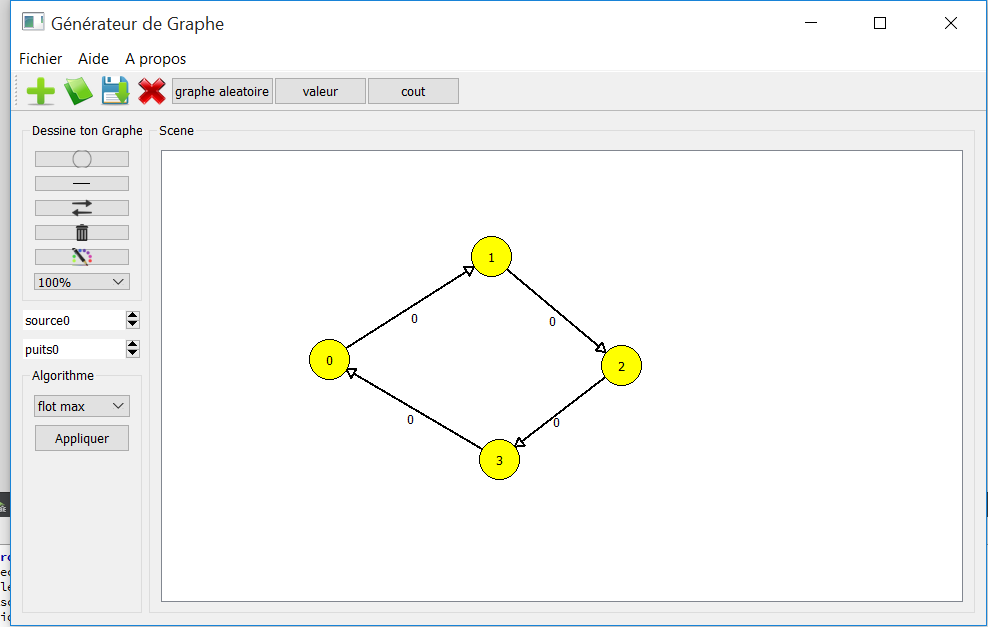
###### ajout d’un arc

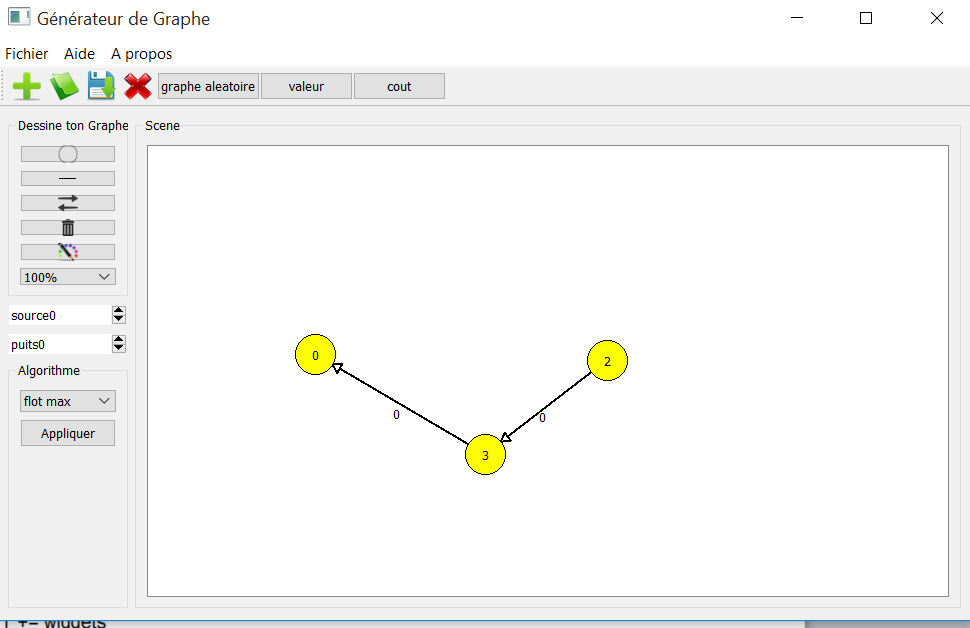


###### supprimer

Pour supprimer un élément, il suffit d’appuyer sur le bouton sélectionner, puis appuyer sur l’élément et ensuite sur le bouton delete.

Par exemple pour supprimer le sommet numéro 1 :

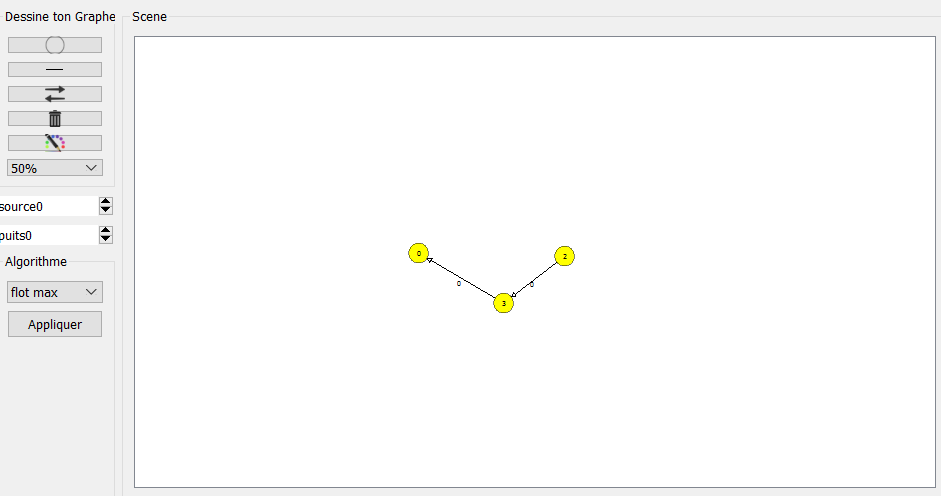




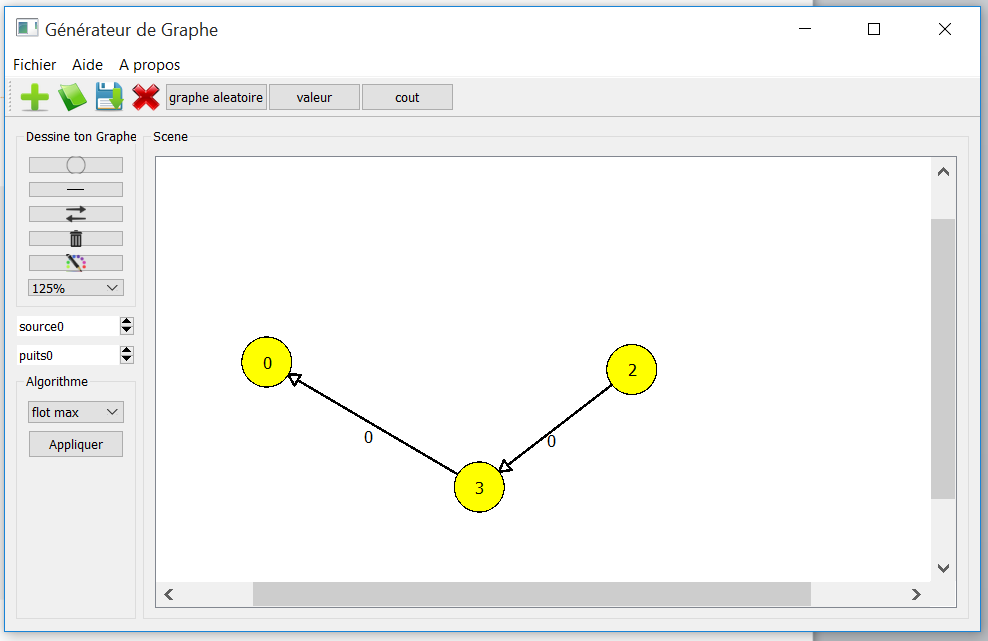
Remarque : Tous les arcs et les arêtes qui sont liées à notre sommet vont être supprimé.

###### zoom

50%

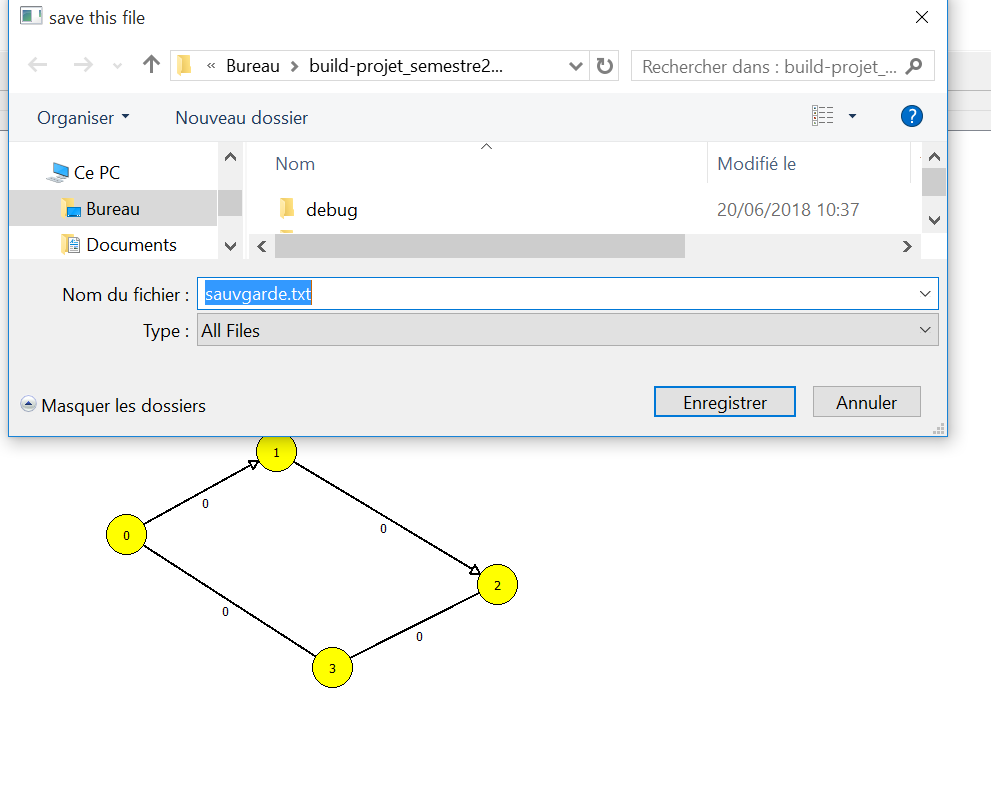


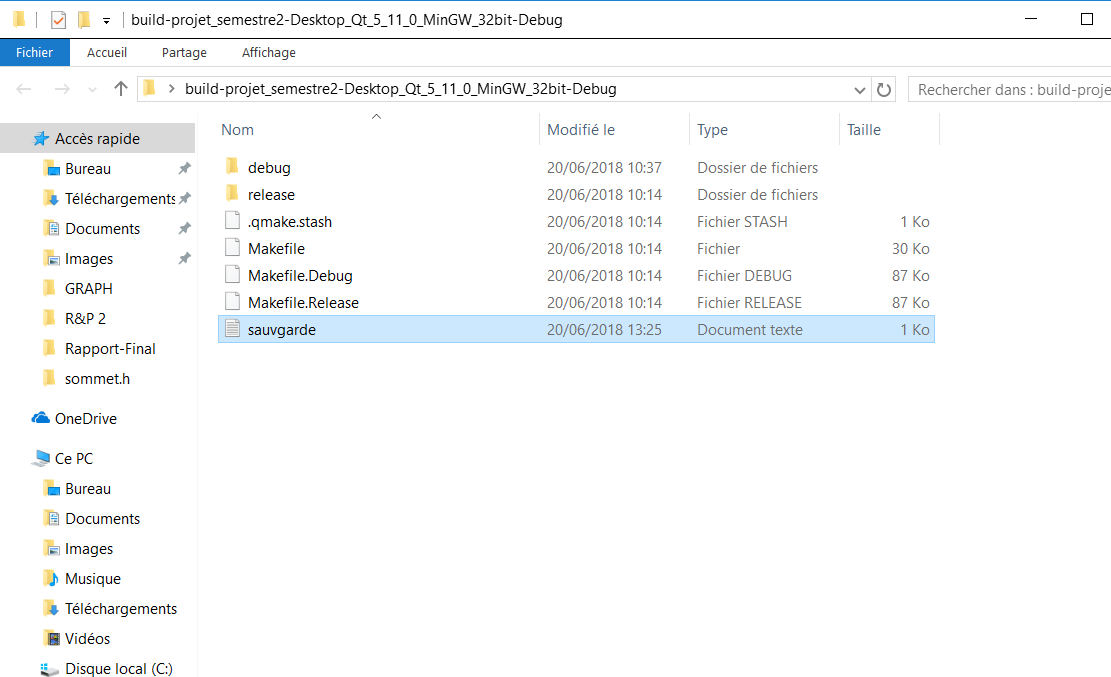
125%



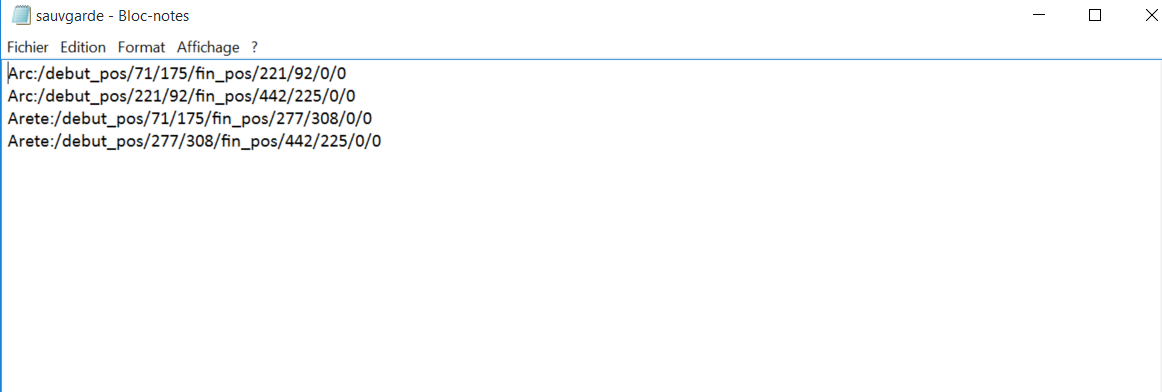
###### pour sauvgarder et ouvrire

Apres avoir créer un graphe il suffit de cliquer sur le bouton save pour le sauvegarder.

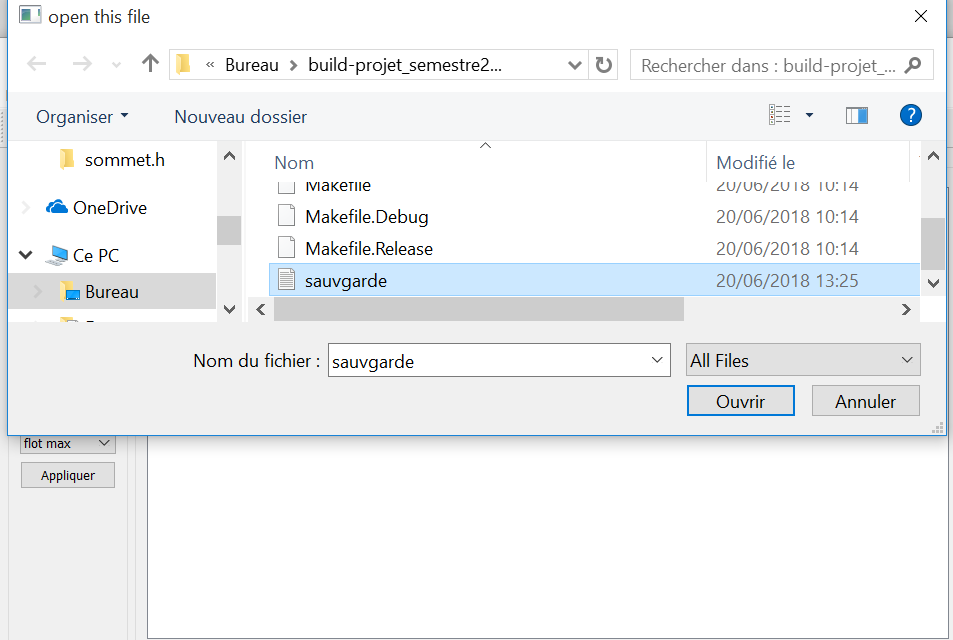




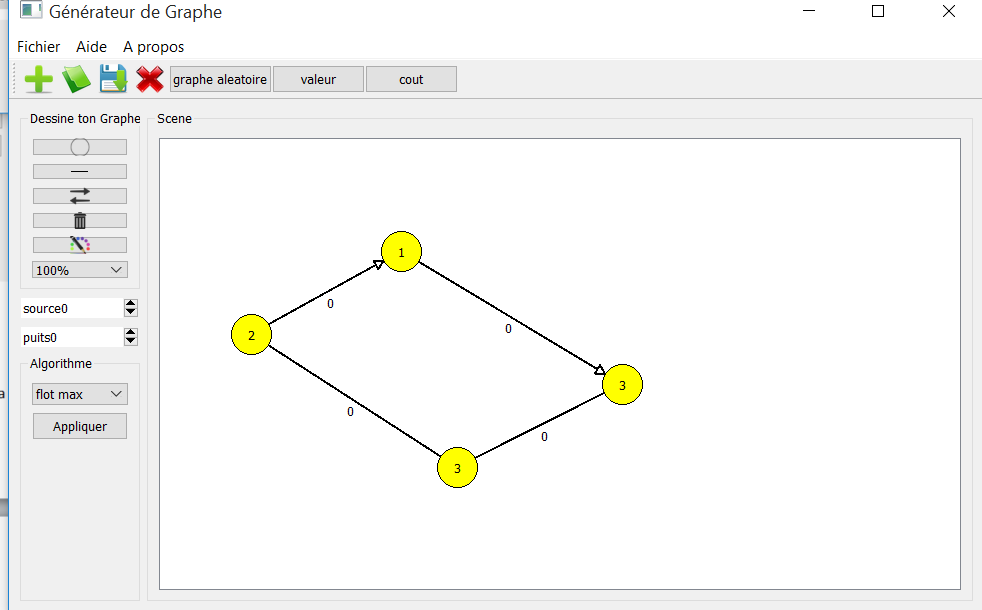
La sauvegarde est sous forme d’un fichier texte qui contient la position du sommet de début et de fin de chaque arc/arête.



##### oucrir une sauvegarde



On appuie sur le bouton open file, l’algorithme traduit le fichier texte en un graphe ,en récupérant les positions des sommets qui relient les arcs/arêtes.

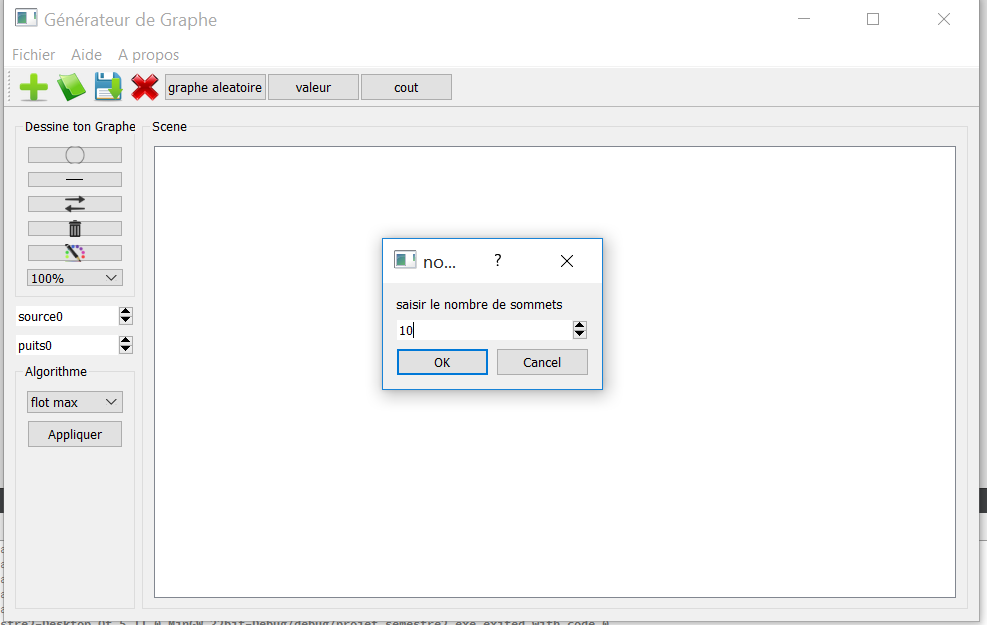


L’algorithme réussit a ouvrir le fichier texte.

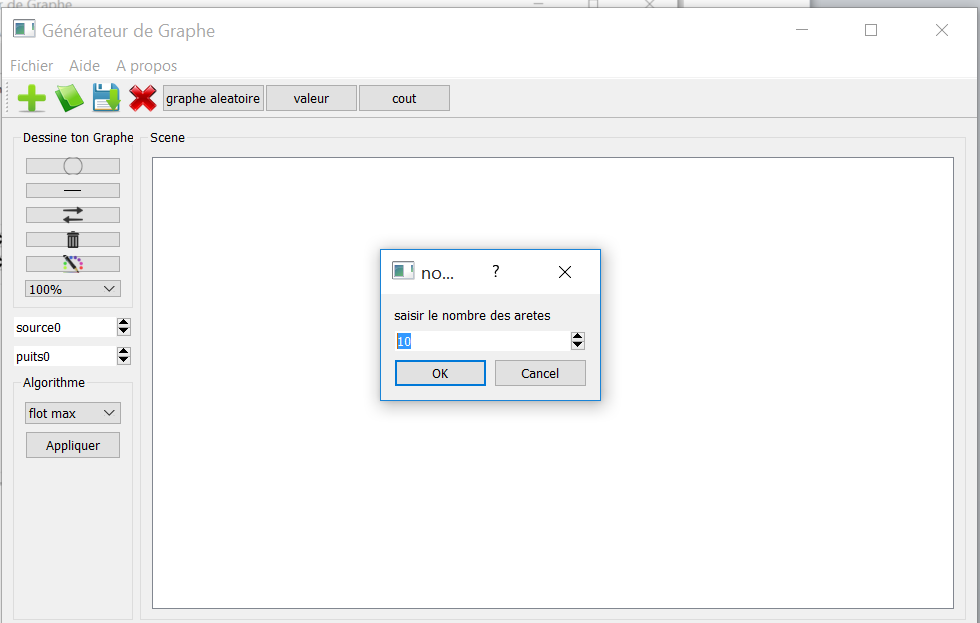
###### graphe aleatoire

On appuie sur le bouton graphe aleatoire ce qui génère l’apparition d’une fenêtre(QInputDialog), on saisit le nombre de sommets, d’arêtes et d’arcs.

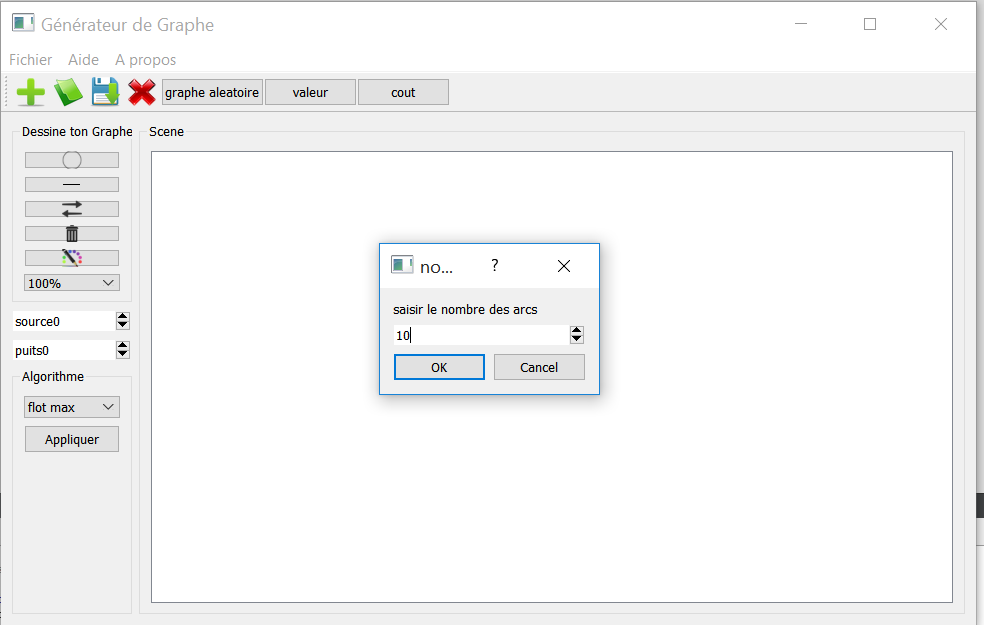
* Nombre de sommets :



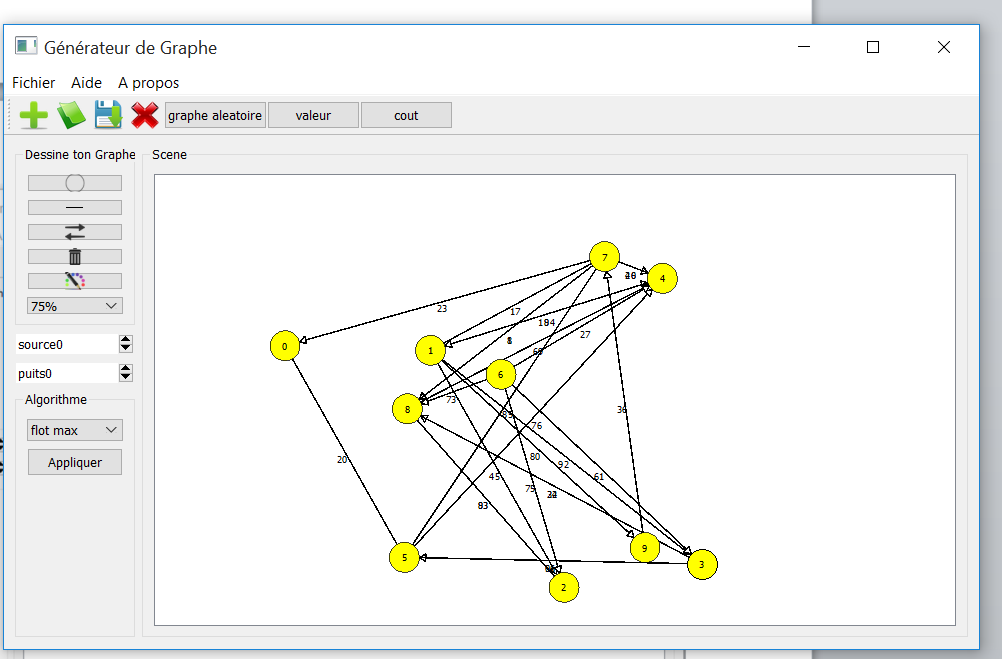
* Nombre d’arêtes :



* Nombre d’arcs :

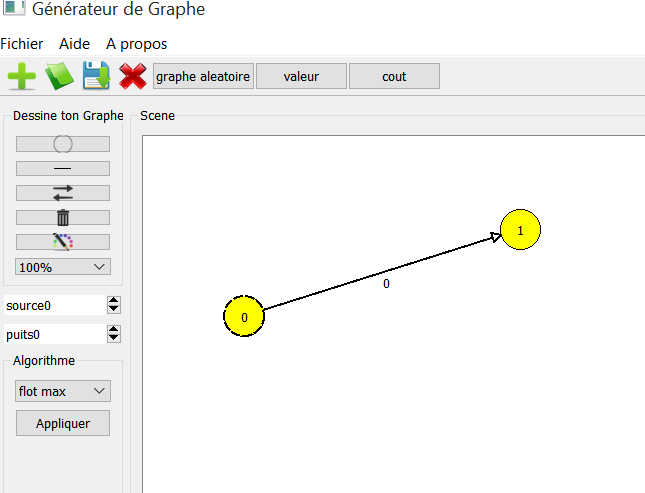


L’algorithmes a reussit a dessiner le graphe :

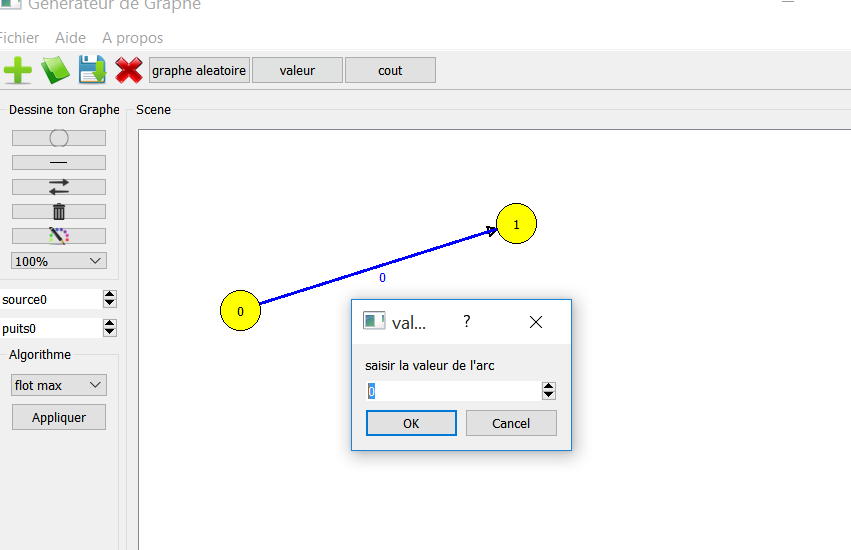


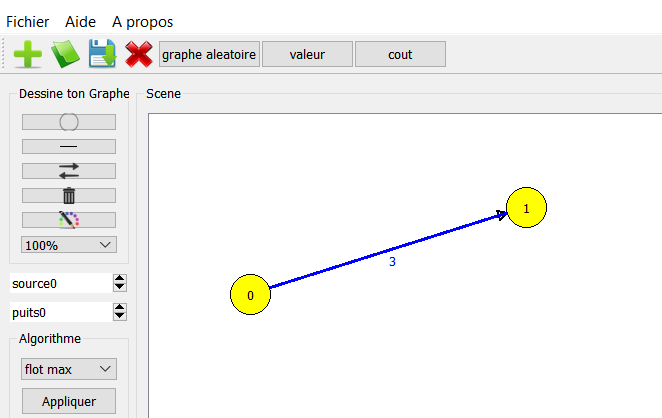
###### ajouter la valeur

Par default la valeur c’est 0

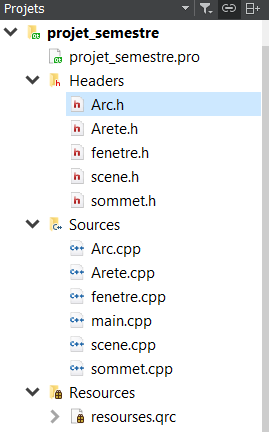


On appuie sur le bouton selectionner puis sur l’arcs et on lui affecte une valeurs



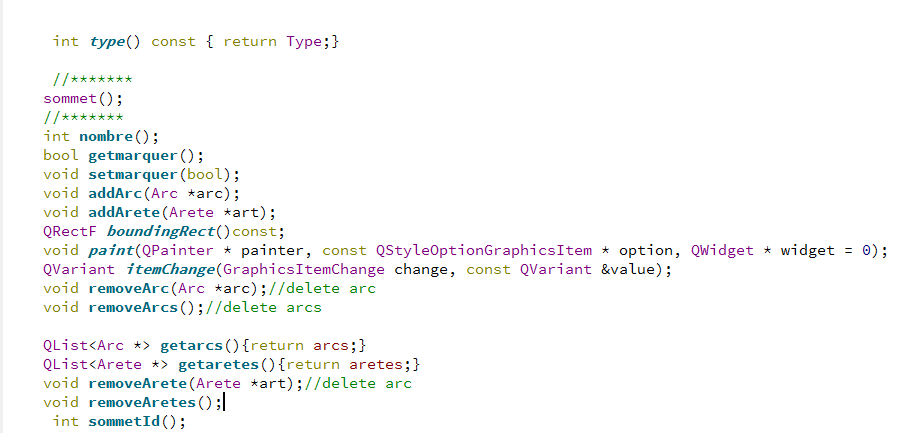


# 4-structure de projet



# 5-Les methodes utilisees dans chaque classe :

##### dans la classe sommet :

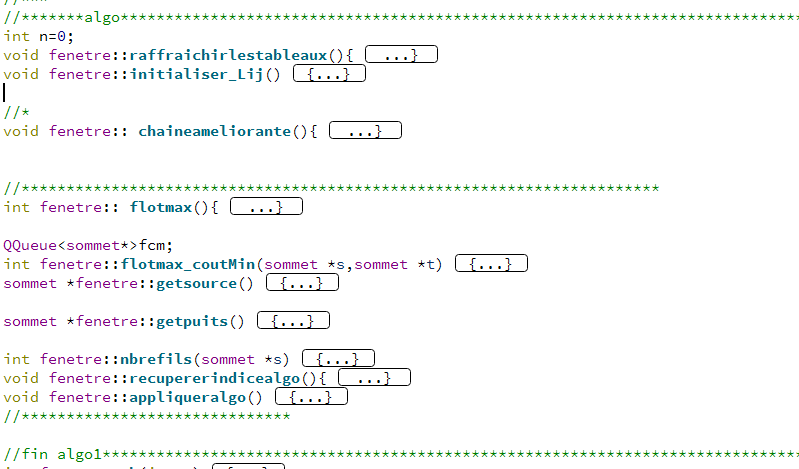


###### dans la classe scene

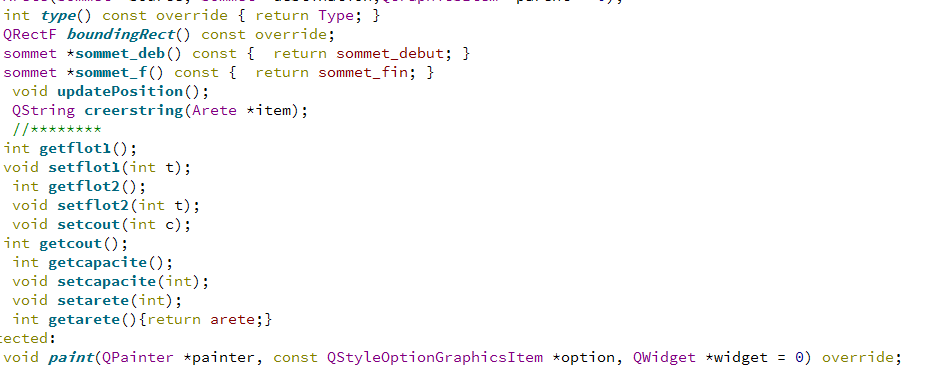


##### dans la classe fenetre





##### dans la classe arete



##### dans la classe arc



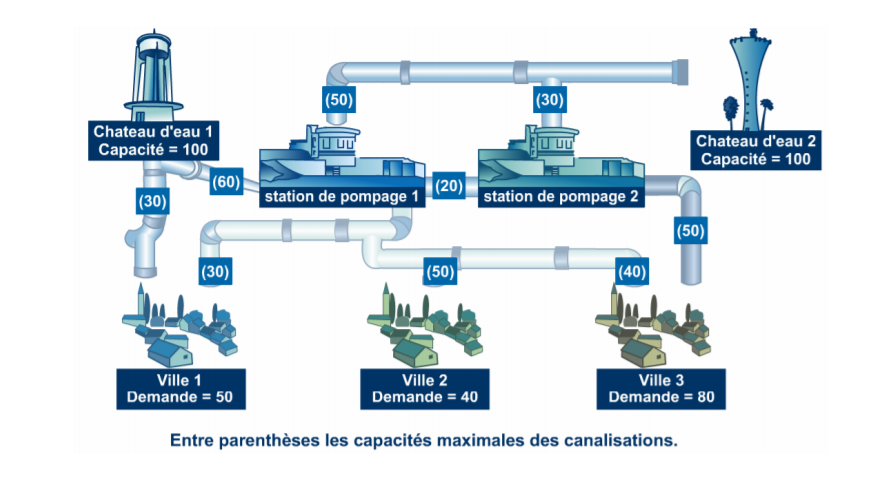
# 6-Generalité et étude fonctionnelle

## DESCRIPTION du Programme:

* Lorsqu’on lance le programme, une fenêtre apparait illustrant un menu qui donne à l’utilisateur la possibilité de dessiner une graphe et d’appliquer un algorithme sur ce dernier.
* Pour générer le programme, soit on le clique manuellement en appuyant sur les boutons des sommets, arcs et arêtes, soit on cliquer le bouton graphe aleatoire et de taper les nombres des éléments nécessaire afin de générer un graphe aleatoire.
* On appuie sur le bouton sélectionner et on sélectionne l’arête qui nous intéresse, puis on appuie sur le bouton ‘valeur’ afin de lui affecter une capacité.
* On désigne le numéro de la source et du puits.
* On choisit l’algorithme qu’ on veut appliquer sur le graphe déjà dessiner.

# 7-Initialisation du problème :

Deux châteaux d'eau alimentent 3 villes à travers un réseau de canalisations au sein duquel se trouvent également des stations de pompage. Les châteaux d'eau ont une capacité limitée qui s'élève pour chacun d'eux à 100 000 m3 .

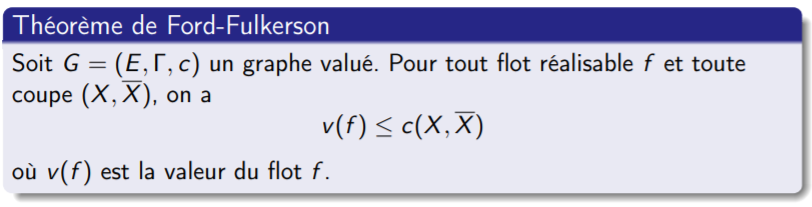


Les villes ont exprimé une demande qui est au minimum de 50 000 pour la ville 1, 40 000 pour la 2 et 80 000 pour la ville 3 en m3 . Les canalisations entre les châteaux d’eau et les villes ont des débits limités. Par exemple, pour la canalisation reliant le château 1 à la ville 1, le débit maximum est de 30 alors que celui de la canalisation reliant la station de pompage 1 à la ville 2 est de 50 en milliers de m3 . Ces valeurs figurent sur le graphique entre parenthèses le long des canalisations. Un premier problème est de déterminer s'il est possible de satisfaire à travers ce réseau la demande des 3 villes et comment ? Pour résoudre ce problème il faut dans un premier temps le modéliser. Pour cela, nous introduisons un nouveau problème standard qui est celui du flot maximal sur un réseau.

# 8- problème de flot :

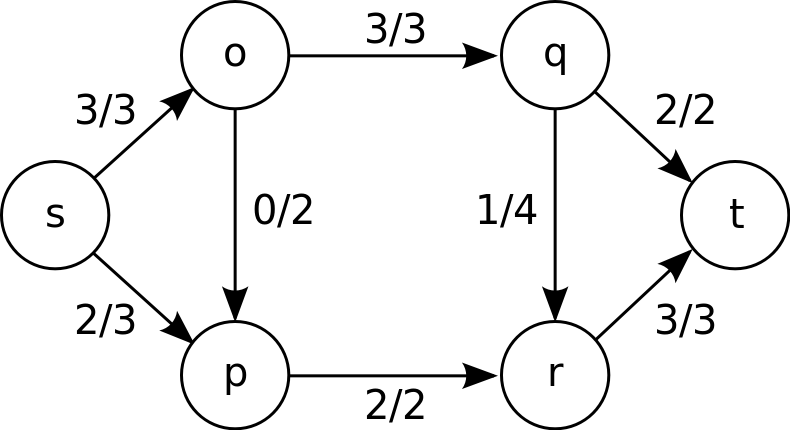
Pour un graphe orienté G(V,E),à chaque arc partant d’un sommet u et arrivant sur un sommet v on associe une valeur dite capacité c(u,v),et on peut définir une fonction dite flot avec les contraintes suivantes f(u,v)<=c(u,v) et f(u,v)=-f(v,u). autre definition : capacité residuelle,reseau residuel,chemin croissant.

### Théorème de Ford-Fulkerson

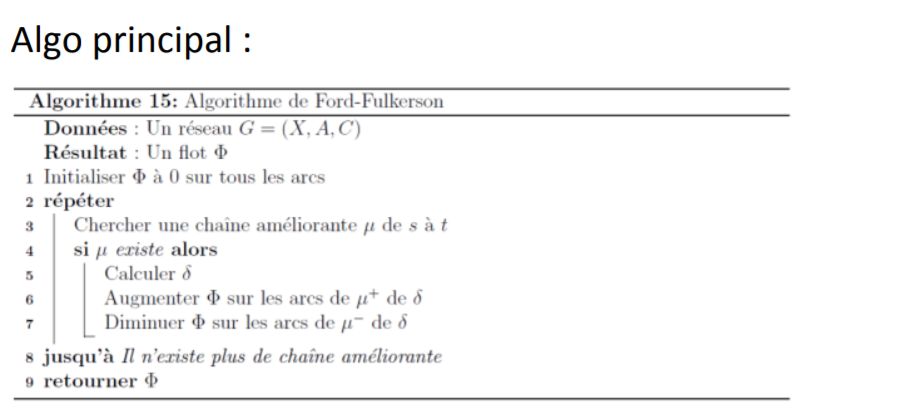


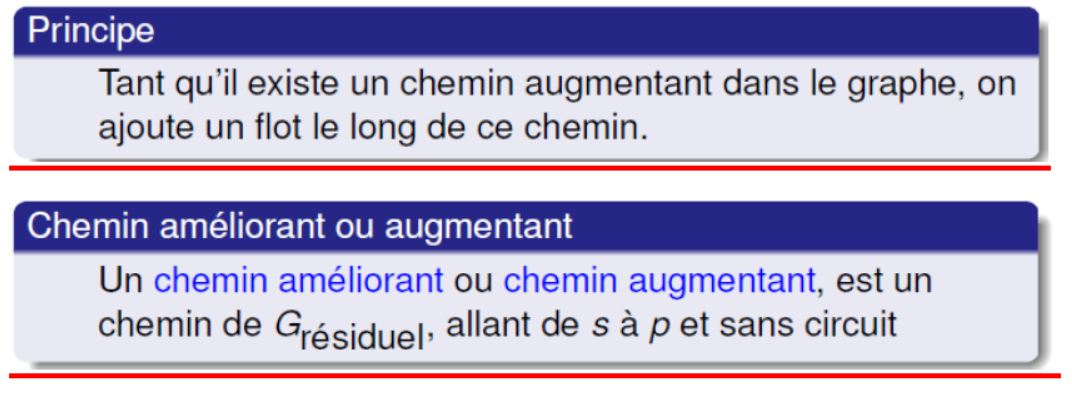
## Flot max

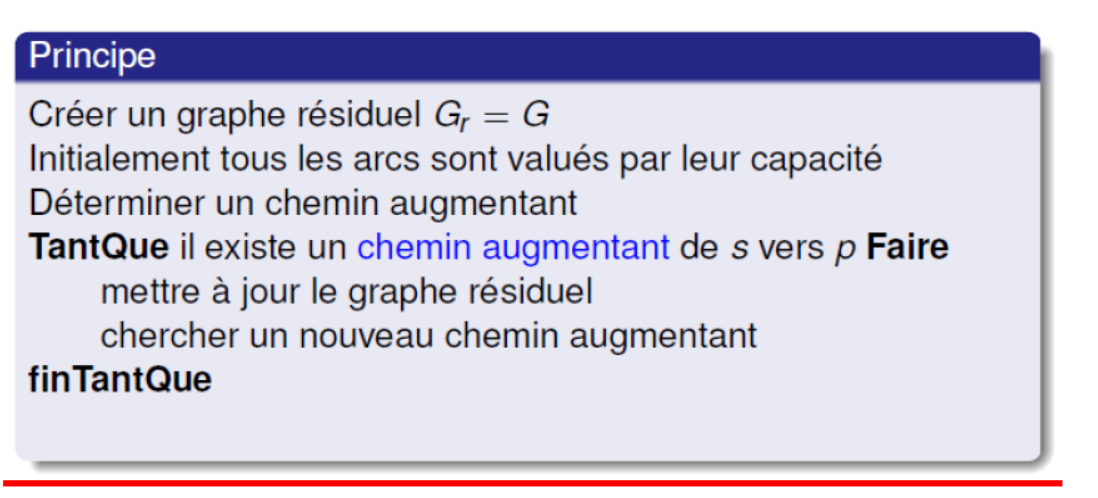
Le problème de flot maximum consiste à trouver, dans un réseau de flot, un flot réalisable depuis une source unique et vers un puits unique qui soit maximum. Ex : On veut par ex. trouver le trafic maximal entre deux villes d’un réseau routier dont on connait la capacité ( nbr de voiture par heure sur chaque tronçon).

******

Algorithme de Ford-Fulkerson :







On construit un graphe résiduel pour pouvoir utiliser l´algo de Ford Fulkerson, qui consiste a :

1/ trouver une chaine améliorante de la source au puits

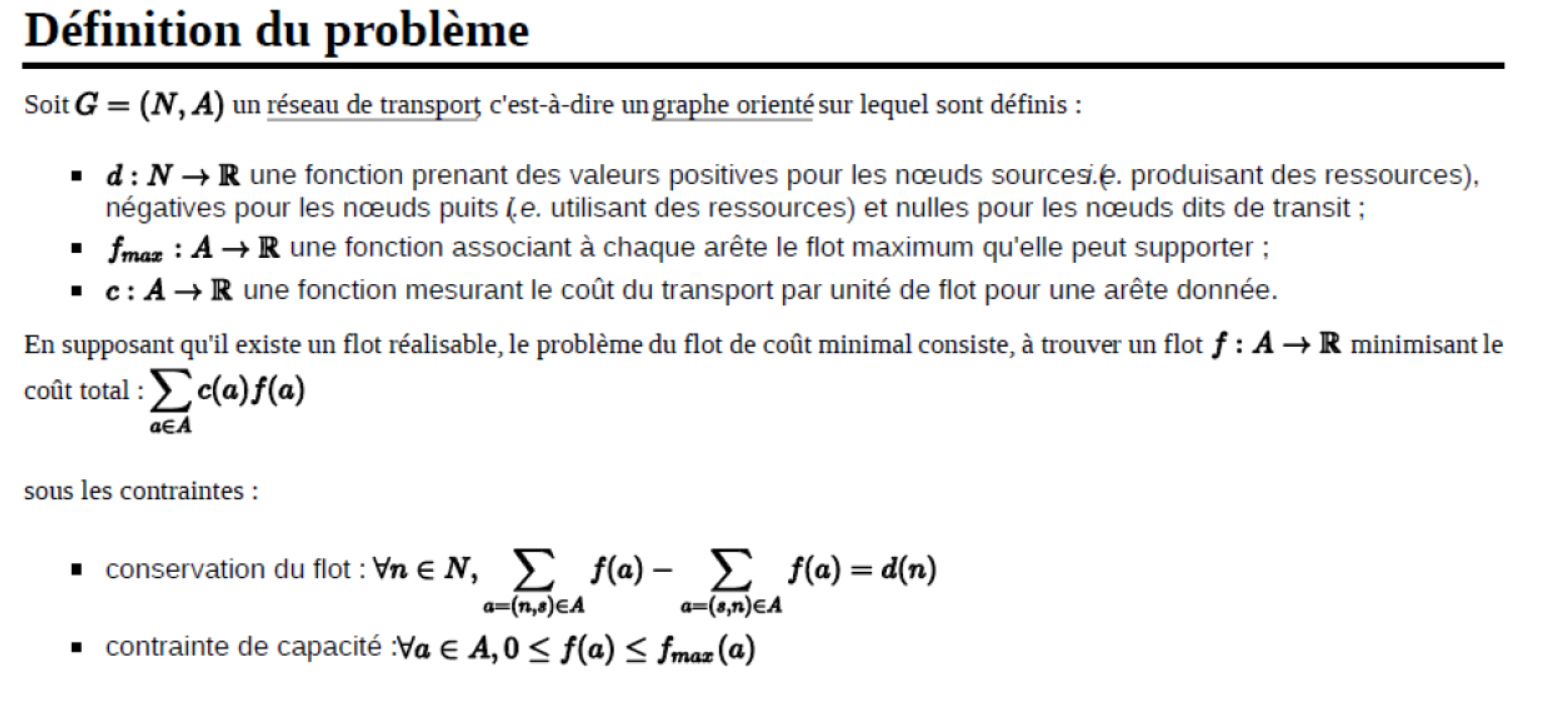
2/ajouter la valeur résiduelle la plus faible sur la chaine améliorante jusqu’ a ce que un arc soit sature

Et en même temps augmenter le flot max

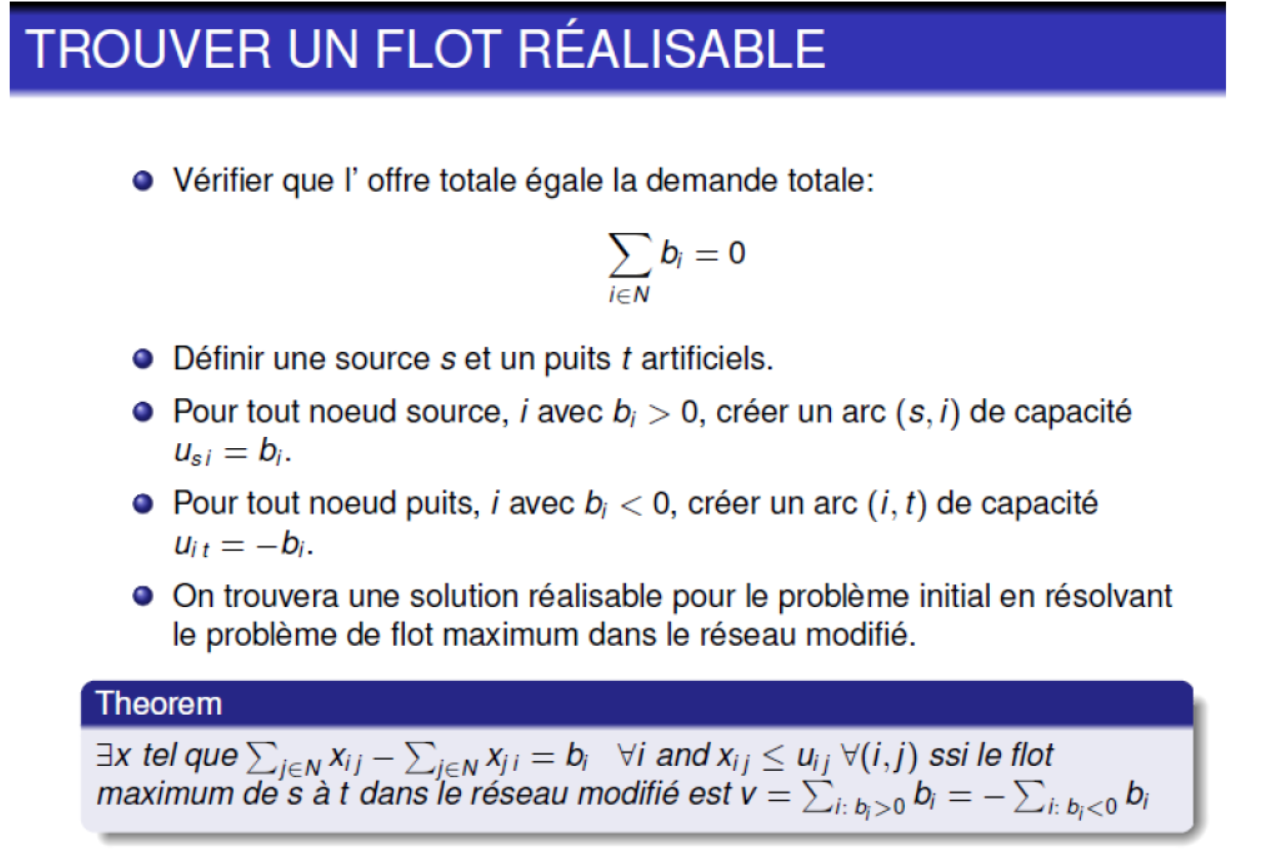
3/on s’arrête dès qu’on trouve plus de chaine améliorante et on retourne le flot max.

## flot max a cout minimum

Definition :

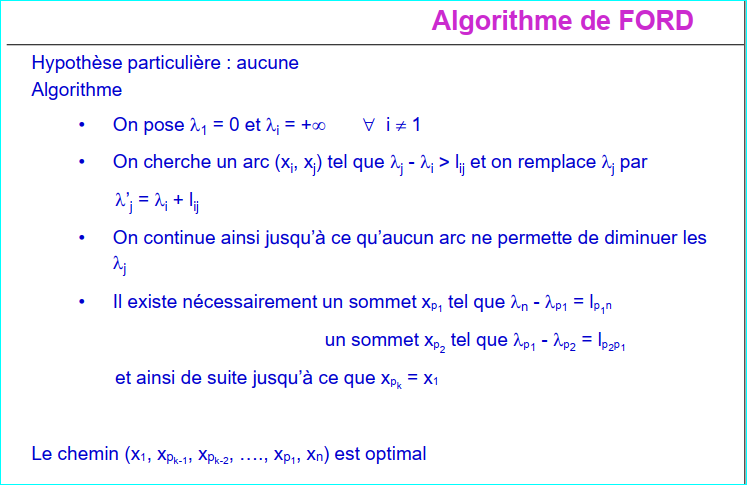


L’algo :



### utilisation de l’algorithme de ford

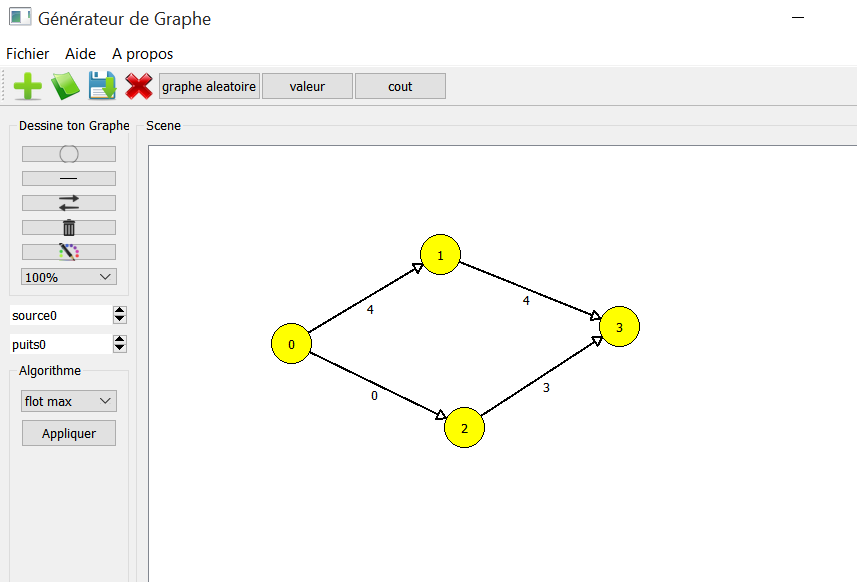
On utilise l’algorithme de Ford afin de trouver le chemin qui a le plus faible cout.



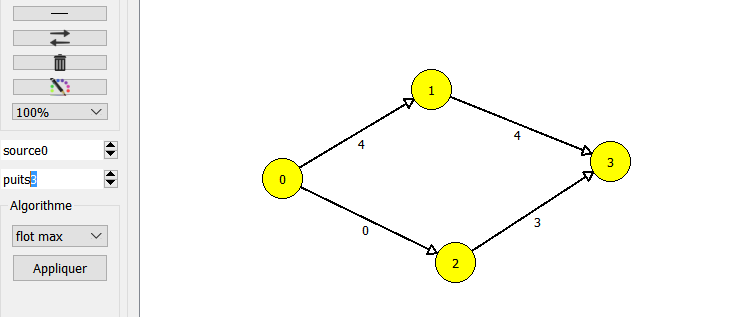
# 9-application des algorithmes :

**Pour l’algorithme de flot max :**

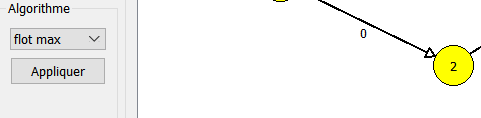
* On construit notre graphe, en lui affecte les valeurs (capacités).



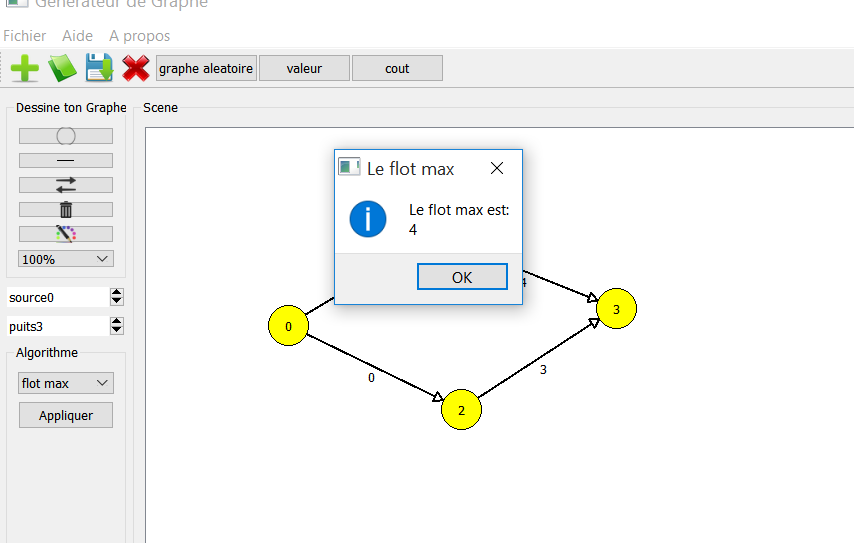
* On choisit la source et le puit.



* On choisit l’algorithme du flot max.

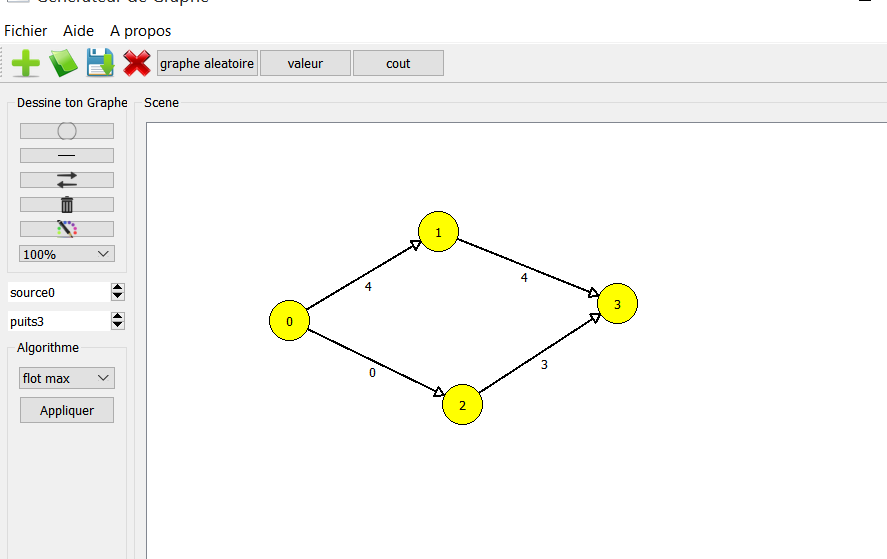


* L’algorithme affiche le résultat.

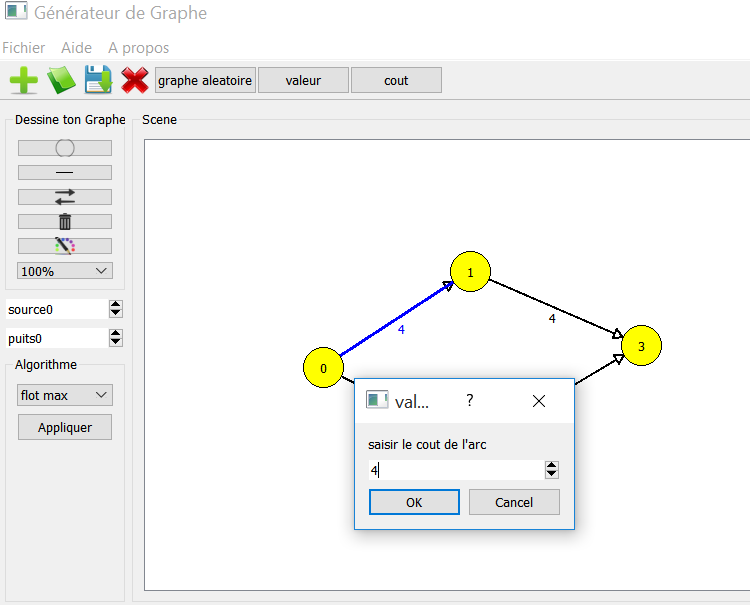


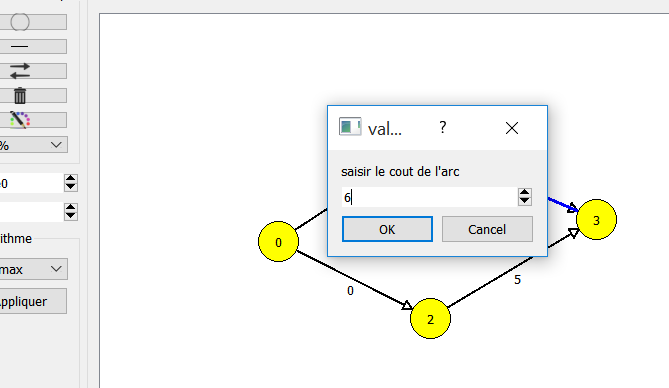
**Pour l’algorithme de flot a cout minimum :**

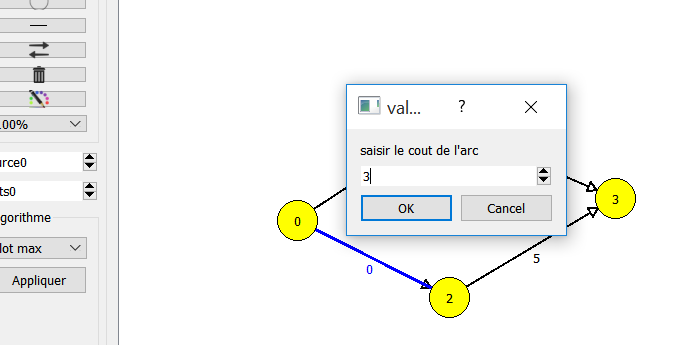
**En prend le même exemple**:

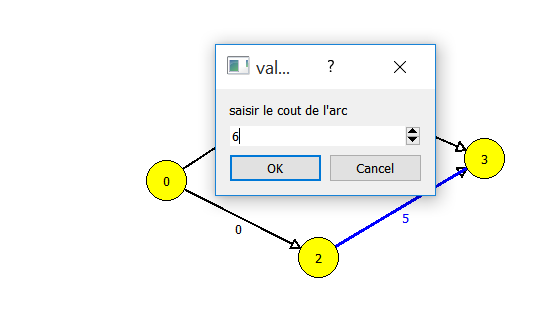


* Il faut ajouter les couts a chaque arc :

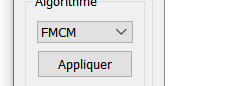




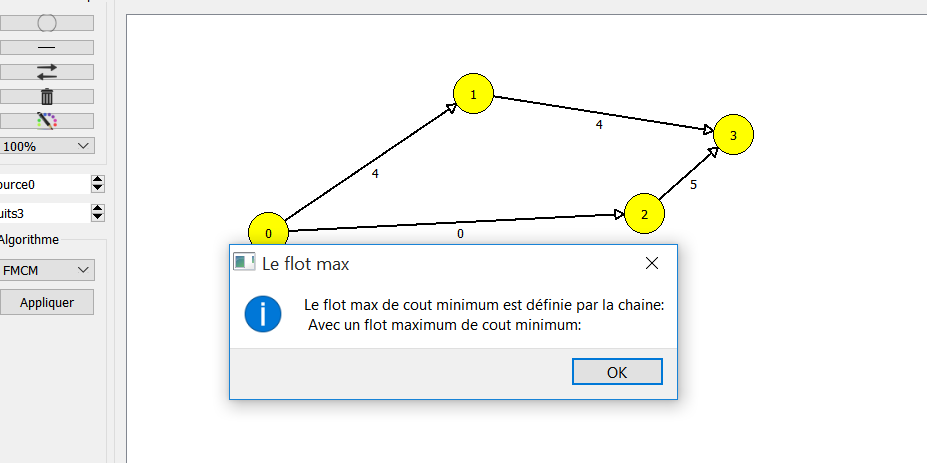




* Apres on choisit le deuxième algorithme.



Problème : l’algorithme n’affiche pas le bon résultat.



Conclusion

Le problème de flot maximal est un problème intéressant qui permet de trouver une répartition efficace des différentes quantités repartis sur un réseau de capacité.

D’autre problèmes ont apparu et ont empreinte le principe de flot maximal à cout minimum.

Depuis l’algorithme de FORD-FULKERSON jusqu’à nos jours, différents algorithmes ont été proposés pour le calcul de flot maximum, ces algorithmes visent, principalement à diminuer la complexité.