

# REV party<sup>1</sup>

Sujet proposé par Vincent Dugat  
Prévu pour une équipe de quatre personnes  
Septembre 2023



## Table des matières

<b>1</b>	<b>Présentation</b>	<b>2</b>
1.1	Election du Burger Suprême . . . . .	2
<b>2</b>	<b>Anonymisation et vérification du vote</b>	<b>3</b>
<b>3</b>	<b>Les méthodes de scrutins</b>	<b>3</b>
3.1	Scrutin uninominal . . . . .	3
3.1.1	Uninominal à un tour . . . . .	3
3.1.2	Uninominal à deux tours . . . . .	3
3.2	Méthodes de Condorcet . . . . .	4
3.2.1	Vainqueur de Condorcet . . . . .	5
3.2.2	Résolution du paradoxe : la méthode du minimax . . . . .	6
3.2.3	Résolution du paradoxe : rangement des paires par ordre décroissant . . . . .	6
3.2.4	Résolution du paradoxe : la méthode de Schulze . . . . .	6
3.3	Jugement majoritaire . . . . .	6

---

1. République Expérimentale Virtuelle

<b>4</b>	<b>Travail à faire : cahier des charges et spécifications</b>	<b>6</b>
4.1	Structures de données . . . . .	7
4.2	Options de la ligne de commande . . . . .	7
4.3	Principe général du programme . . . . .	8
4.4	Liste ordonnée . . . . .	8
4.5	Liste des modules à programmer . . . . .	9
4.6	Evaluation des modules . . . . .	9
4.6.1	Les tests (rappels) . . . . .	9
4.6.2	Documentation . . . . .	10
<b>5</b>	<b>Annexes</b>	<b>11</b>
5.1	Format CSV . . . . .	11
5.2	Le programme de simulation de votes . . . . .	11
5.3	Graphes orientés et graphe de tournoi . . . . .	11

---

Après une présentation générale du projet, vous trouverez la partie intitulée "Travail à faire : cahier des charges et spécifications" qui précise exactement ce qu'il faut programmer. Prenez le temps de bien lire cette partie. Le respect des spécifications sera pris en compte dans l'évaluation. Certains aspects sont (involontairement) flous. Ça n'est pas si rare dans un cahier des charges. Vous devez demander l'interprétation de ces points à votre tuteur.

## 1 Présentation

Des pays, entreprises, organisations diverses recourent au vote pour désigner des délégués, représentants, dirigeants, faire des choix divers. Il existe de nombreux modes de scrutins différents. Ces différences portent sur les modes opératoires mais aussi sur les protocoles et modes de calculs. Ces deux derniers points ont fait, et font toujours, l'objet de travaux mathématiques visant à en cerner les propriétés et conséquences.

Au final, les modes opératoires et modes de calculs des scrutins ne sont rien de plus que des algorithmes que nous allons implémenter en langage C.

Vous êtes invité à lire le document suivant à titre d'exemple introductif :

<http://images.math.cnrs.fr/La-democratie-objet-d-etude.html>

### 1.1 Election du Burger Suprême

Afin d'avoir de la matière pour nos calculs qui ne soit pas limités à des jeux de données de tests, nous allons commencer par organiser un vote au moyen d'un sondage Moodle. C'est un vote "pour jouer", mais il sera organisé comme un vrai vote. Le prétexte est d'élire votre «burger préféré».

Le sondage consiste à classer les dix candidats selon une échelle de Likert en classant par ordre de préférence de 1 (le préféré) à 10 (le moins aimé). Il peut y avoir des ex-quo.

Une fois le sondage clos, la base de données des réponses (votes) peut être exportée (par le staff) au format csv (comma separated values). En fait, Moodle sépare les données avec des "tab".

Le fichier débarrassé des informations inutiles et anonymisé sera alors proposé au téléchargement sur Moodle. Ce fichier contenant les ballots du vote par classement préférentiel sera la donnée de vos programmes. C'est donc vos programmes qui vont dire qui gagne l'élection selon les différentes méthodes.

Un deuxième tour pour le vote uninominal à deux tours sera également organisé.

Il est évident que dans un cas réel ces manipulations du fichier de ballots posent des problèmes de sécurité et d'intégrité du vote. Ces questions sont résumées dans un document annexe pour ceux que cela intéresse.

## 2 Anonymisation et vérification du vote

Pour que le résultat d'un vote soit accepté par les votants et les candidats, on doit respecter certains principes :

1. seuls les inscrits sur les listes électorales peuvent voter, et ils votent une seule fois
2. La procédure de vote doit être expliquée précisément (rien de flou et rien de caché)
3. Le vote est confidentiel (nul ne doit craindre de représaille pour son choix)
4. Tout le monde doit pouvoir vérifier l'intégrité du vote
5. Le dépouillement et les calculs doivent être publiques et transparents

Nous allons nous focaliser ici sur la confidentialité du vote et la vérification de son vote par le votant, c'est à dire le point 3 de la liste ci-dessus. Nous allons nous inspirer de la méthode utilisée par Debian pour l'élection de son chef de projet.

Au moment du vote, le votant reçoit un code aléatoire qu'il est le seul à connaître. Son vote est enregistré dans le fichier avec le hashé du code. Il suffit au votant de hasher le code pour trouver dans le fichier la ligne correspondant à son vote et le vérifier.

Pour le vote du «Burger Suprême qui les gouverne tous», le fichier de vote sera anonymisé avec cette méthode et vous recevrez par email votre code (si l'IRIT m'autorise à faire un envoi de masse). Le fichier anonymisé sera ensuite mis en ligne.

## 3 Les méthodes de scrutins

### 3.1 Scrutin uninominal

Ces méthodes, comme leur nom l'indique, demande de ne voter que pour une seule personne. S'il y a ex aequo on utilise le privilège de l'âge (le plus âgé est élu).

#### 3.1.1 Uninominal à un tour

C'est la méthode la plus simple à mettre en oeuvre. C'est celle qui est utilisée pour l'élection de vos délégués de groupes de CTD et de TP, mais aussi pour les élections locales au Royaume Uni et dans d'autres pays du Commonwealth.

Principe : chaque électeur vote pour un et un seul candidat. A la fin du vote, on compte le nombre de voix de chacun des candidats. Celui qui a obtenu le plus grand nombre de voix l'emporte.

Avantages : très simple à mettre en oeuvre, peu de calculs.

Inconvénients : un candidat peut être élu avec un pourcentage assez faible des suffrages, ce qui questionne sa légitimité. De plus il risque d'avoir contre lui une majorité de mécontent et ce dès le début.

#### 3.1.2 Uninominal à deux tours

C'est le système qu'on emploie en France pour, entre autre, élire le président de la république. Comme précédemment, chaque électeur vote pour un et un seul candidat. A la fin du vote, on compte le nombre de voix de chacun des candidats. Les deux candidats qui ont eu les plus grands nombres de voix restent en compétition pour un second tour (sauf si un candidat a obtenu plus de 50% des voix auquel cas il est élu). A l'issue du second tour, le candidat qui a obtenu le plus de voix l'emporte.

Avantages : plus de légitimité pour celui qui est élu

Inconvénients : crée des phénomènes de vote utile, de stratégie de reports de voix et peut favoriser l'abstention au second tour, et donc poser là aussi des problèmes de légitimité.

## 3.2 Méthodes de Condorcet

Dans son *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, le marquis de Condorcet, mathématicien et philosophe français du XVIIIe siècle, met en évidence le fait que le vote à la pluralité (uninominal à un tour) peut très bien ne pas représenter les désirs des électeurs.

Il proposa une méthode, dite méthode de Condorcet (aussi appelée scrutin de Condorcet ou vote Condorcet) qui est un système de vote dans lequel l'unique vainqueur, s'il existe, est le candidat qui, comparé tour à tour à tous les autres candidats, s'avère à chaque fois être le candidat préféré, c'est-à-dire le vainqueur de Condorcet.

Un scrutin de Condorcet obéit au principe de Condorcet, qui se veut être le principe démocratique le plus naturel possible, à savoir que « si un choix est préféré à tout autre par une majorité, alors ce choix doit être élu. » En effet, ce principe semble logique dans le cadre d'un scrutin démocratique. Il signifie autrement dit que, lors d'une élection, si une majorité d'électeurs préfère un candidat à tous les autres candidats, alors ce candidat préféré est celui qui doit être désigné vainqueur du scrutin.

Rien ne garantit la présence d'un candidat satisfaisant au critère de victoire. Ainsi, tout système de vote fondé sur la méthode comparative de Condorcet doit prévoir un moyen de résoudre les votes pour lesquels ce candidat idéal n'existe pas.

L'option (ou le candidat) désigné vainqueur par la méthode de Condorcet est appelé *vainqueur de Condorcet*.

Le lien suivant vous donne des exemples concrets.

[https://fr.wikipedia.org/wiki/M%C3%A9thode\\_de\\_Condorcet](https://fr.wikipedia.org/wiki/M%C3%A9thode_de_Condorcet)

On peut représenter ces résultats avec une matrice des duels (en reprenant l'exemple de Wikipedia) :

	A	B	C
A	0	25	23
B	35	0	19
C	37	41	0

Cette matrice à son tour peut être représentée par un graphe orienté dont les candidats sont les sommets. Un arc  $(a, b)$  où  $a$  et  $b$  sont des candidats signifie  $a$  est préféré à  $b$  ( $a$  domine  $b$  dans le langage des graphes). On peut de plus associer à cet arc le pourcentage des préférences ou les nombres de voix. L'arc devient un arc étiqueté  $(a, b, p)$  où  $p$  est un nombre.

On remarque qu'on ne fait figurer dans la graphe que les arcs gagnants. Ce graphe est le plus souvent un graphe de tournoi (cf. annexe).

### Déroulement du vote

Chaque électeur classe les candidats par ordre de préférence. Ce vote s'appelle un "ballot".

### Décompte des votes

Le dépouillement du scrutin consiste à simuler l'ensemble des duels possibles : pour chaque paire de candidats, on détermine le nombre d'électeurs ayant voté pour l'un ou l'autre en vérifiant, sur chaque bulletin de vote, comment l'un était classé par rapport à l'autre. Ainsi pour chaque duel, il y a un candidat vainqueur. Dans la plupart des cas, il y a un unique candidat qui remporte tous ses duels : il s'agit du vainqueur du scrutin. La section suivante décrit ce qui se passe dans les rares cas où aucun candidat ne remporte tous ses duels.

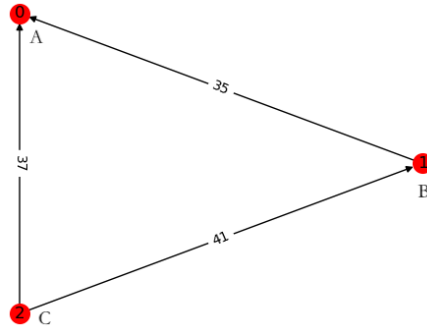


FIGURE 1 – Graphe de l'exemple théorique.

### 3.2.1 Vainqueur de Condorcet

**Définition :** Si, parmi les candidats à une élection, il en existe un qui, face à n'importe quel autre, lui est préféré par une majorité d'électeurs, alors ce candidat est appelé le vainqueur de Condorcet.

**Définition :** On dit qu'une méthode électorale satisfait le critère de Condorcet quand, lorsqu'il y a un vainqueur de Condorcet, c'est toujours lui que cette méthode déclare vainqueur (sous réserve que les électeurs aient voté selon leurs préférences véritables).

**Théorème :** Il ne peut pas y avoir plus d'un vainqueur de Condorcet à une élection.

Dans le graphe des duels un vainqueur de Condorcet aura la particularité de n'avoir que des arcs sortants. On dit qu'il domine tous les autres sommets. S'il y a un conflit alors aucun sommet ne domine tous les autres et le graphe comporte un circuit.

Le site ci-dessous fait des simulations intéressantes :

<https://jorisdeguet.github.io/PrefVote/tideman.html>

Les méthodes proposées dans les sections qui suivent, proposent des solutions pour trouver un vainqueur dans la cas où il y a un conflit.

**Avantages :** les électeurs ont plus tendance à voter selon leurs vraies préférences. Le vainqueur a la légitimité de la préférence majoritaire sur tous les autres candidats.

**Inconvénients :** plus difficile à mettre en place et à dépouiller.

#### Résolution des conflits

Il arrive qu'aucun candidat ne soit élu à la suite du décompte des votes : Condorcet avait remarqué cet important paradoxe inhérent à la méthode, appelée le paradoxe de Condorcet : dans une élection dont les candidats sont  $A$ ,  $B$  et  $C$ , dès que le nombre des électeurs est supérieur à deux,  $A$  peut être préféré à  $B$ , lui-même préféré à  $C$ , lui-même préféré à  $A$ . Plusieurs méthodes sont utilisables pour résoudre ce conflit de circularité.

**Remarque :** S'il y a un vainqueur de Condorcet, auquel cas la méthode de résolution de conflit n'est pas appliquée et le vainqueur du scrutin sera le vainqueur de Condorcet. S'il y a un conflit, une méthode de résolution de conflit est appliquée.

### 3.2.2 Résolution du paradoxe : la méthode du minimax

Le protocole est le suivant : pour chacun des candidats, on regarde le pire des scores qu'il ferait dans ses différents faces-à-faces, puis on déclare élu celui des candidats dont le pire des scores est le meilleur. Cette méthode est appelée méthode minimax, car elle regarde qui est le meilleur (« max ») dans la situation où il est le moins bon (« mini »).

Exemple :

<http://images.math.cnrs.fr/Et-le-vainqueur-du-second-tour-est.html>

**Inconvénients** : peut être biaisée par des stratégies entre partis.

### 3.2.3 Résolution du paradoxe : rangement des paires par ordre décroissant

[fr.wikipedia.org/wiki/Méthode\\_Condorcet\\_avec\\_rangement\\_des\\_paires\\_par\\_ordre\\_décroissant](http://fr.wikipedia.org/wiki/Méthode_Condorcet_avec_rangement_des_paires_par_ordre_décroissant)

Pour cette méthode lisez la page Wikipédia qui en parle très bien.

La recherche des circuits du graphe se fera grâce à un algorithme de recherche en profondeur modifié. Il correspond à la librairie à télécharger sur Moodle. Vous n'avez donc pas à le programmer.

### 3.2.4 Résolution du paradoxe : la méthode de Schulze

Les pages Wikipedia ci-dessous sont suffisantes pour comprendre le principe :

[https://en.wikipedia.org/wiki/Schulze\\_method](https://en.wikipedia.org/wiki/Schulze_method)

[https://fr.wikipedia.org/wiki/Méthode\\_Schulze](https://fr.wikipedia.org/wiki/Méthode_Schulze)

## 3.3 Jugement majoritaire

Pour finir une méthode différente de celles de Condorcet : le jugement majoritaire qui est une méthode de vote par valeurs (les électeurs attribuent une mention à chaque candidat et peuvent attribuer la même mention à plusieurs candidats) pour laquelle la détermination du gagnant se fait par la médiane plutôt que par la moyenne. Nous aurons six mentions : très bien (TB), bien (B), assez bien (AB), passable (P), médiocre (M), à fuir.

[https://fr.wikipedia.org/wiki/Jugement\\_majoritaire](https://fr.wikipedia.org/wiki/Jugement_majoritaire)

Pour utiliser le même fichier de vote que pour les méthodes de Condorcet, nous utiliserons la correspondance du tableau ci-dessous :

TB	B	AB	P	M	A fuir
1	2-3	4-5	6-7	8-9	10

## 4 Travail à faire : cahier des charges et spécifications

Moodle permet de récupérer le résultat d'un sondage (le vote) sous forme d'un fichier CSV contenant les ballots et d'autres informations. Le programme devra construire les structures de données à partir de ce fichier.

### 4.1 Structures de données

Tous les tableaux employés dans le programme devront être dynamiques. Leur taille sera allouée à la demande avec éventuellement des *realloc*. Le fichier *global.h* donne des exemples de structures de données. Ce ne sont que des exemples, à vous de voir si vous les garder ou pas.

## 4.2 Options de la ligne de commande

Le programme doit récupérer les options de la ligne de commande. Ces options sont normalisées sous la forme d'une balise : un tiret "-" suivit d'une lettre, et d'un paramètre de la balise qui est une chaîne de caractères.

Vous pouvez utiliser le header getopt.h (`#include <getopt.h>`) pour faciliter l'analyse des paramètres de la ligne de commande.

Balise	Paramètre	Action
-i	nom de fichier csv	traite le fichier cité
-d	nom de fichier csv	traite le fichier cité
-o <a href="#">optionnel</a>	nom de fichier txt	fichier de log
-m	méthode	précise la méthode de scrutin à utiliser

où méthode  $\in \{uni1, uni2, cm, cp, cs, jm, all\}$  avec :

paramètre	méthode associée
uni1	uninomiale à un tour
uni2	uninomiale à deux tours
cm	méthode de Condorcet et minimax
cp	méthode de Condorcet et rangement des paires
cs	méthode de Condorcet et méthode de Schulze
jm	méthode du jugement majoritaire
all	toutes les méthodes

Dans l'analyse des options : Si la balise -i est présente, le fichier csv est le résultat d'un vote par classement (ensembles de ballots).

Si la balise -d est présente à la place de -i alors le fichier csv argument est une matrice de duels.

**Attention :** Les deux options -i et -d sont incompatibles mais l'une des deux doit être présente.

Si la balise -o <nom>.txt est donnée, le fichier <nom>.txt sera considéré comme un fichier de log. Une variable logfp pourra être affectée au résultat de l'ouverture de ce fichier, sinon logfp= stdout. Ce fichier (ou à défaut l'écran) recevra les affichages intermédiaires permettant la vérification du vote (le suivi des calculs).

La balise -m suivie d'un nom de méthode  $\in \{uni1, uni2, cm, cp, cs, jm, all\}$  permet d'appliquer une méthode de scrutin. La méthode de Condorcet correspond aux paramètres :  $\{cm, cp, cs\}$ . Le paramètre *all* permet d'appliquer toutes les méthodes valides l'une après l'autre et affiche le résultat de chacune

Si l'option -d est présente, elle désactive l'emploi des argument *uni1* et *uni2* de l'option -m (Ces méthodes ont besoin des ballots).

En cas d'erreur : balise inconnue, paramètre de balise incorrect ou absent, balises incompatibles, fichier inexistant ou mal formé, le programme affichera un message d'erreur et sortira avec un "exit(EXIT\_FAILURE)".

**Les balises pourront être mises dans un ordre quelconque.**

## 4.3 Principe général du programme

La méthode générale est la suivante :

— Analyse des balises.

- Lire le fichier csv et remplir une matrice de chaînes de type `t_mat_char_star_dyn` (pour  $-i$  et  $-d$ ). On conserve la ligne des entêtes.
- Créer la matrice des duels à partir du tableau précédent (le traitement tiendra compte de l'option  $-i$  ou  $-d$ ). Le type est `t_mat_int_dyn`.
- Créer la liste des arcs correspondants à la matrice (on crée le graphe).
- Appliquer la méthode de scrutin sur la matrice des duels ou sur le graphe suivant les méthodes (l'un ou l'autre sont plus naturels suivant les méthodes).
- Afficher les résultats.

Votre programme sera compilé au moyen d'un *Makefile* et l'exécutable portera le nom de *scrutin*.

L'affichage des résultats sera normalisé suivant la forme :

*Mode de scrutin* : `<nom de la méthode>, <nb> candidats, <nb>, votants, vainqueur = <nom du vainqueur>`

Dans le cas des scrutins uninominaux on affichera le score en plus :

*Mode de scrutin* : `<nom de la méthode>, <nb> candidats, <nb>, votants, vainqueur = <nom du vainqueur>, score = <nb>%`

Pour simplifier les affichages et utiliser le même sous-programme, le scrutin uninominal à deux tours, fera deux affichages pour le tour 1 (un pour chacun des deux candidats de tête), et un affichage pour le second tour.

**Remarque :** Pour le deuxième tour du vote uninominal à deux tours, si les deux candidats sélectionnés au premier tour sont *A* et *B*, nous pouvons faire un deuxième tour virtuel en supposant que ceux qui ont préféré *A* à *B*, au premier tour, vont voter *A* au second tour, et même chose pour *B*.

Ce résultat pourra être comparé à celui d'un deuxième tour réel qui revient à une méthode uninominale à un tour avec deux candidats.

#### Exemple :

```
Mode de scrutin : uninominal à un tour, 10 candidats, 100 votants, vainqueur = Astérix, score = 51%
Mode de scrutin : uninominal à deux tours, tour 1, 10 candidats, 100 votants, vainqueur = Astérix, score = 51%
Mode de scrutin : uninominal à deux tours, tour 1, 10 candidats, 100 votants, vainqueur = Obélix, score = 49%
Mode de scrutin : uninominal à deux tours, tour 2, 2 candidats, 100 votants, vainqueur = Astérix, score = 51%
Mode de scrutin : Condorcet minimax, 10 candidats, 100 votants, vainqueur = Astérix
Mode de scrutin : Condorcet paires, 10 candidats, 100 votants, vainqueur = Astérix
Mode de scrutin : Condorcet Schulze, 10 candidats, 100 votants, vainqueur = Astérix
Mode de scrutin : Jugement majoritaire, 10 candidats, 100 votants, vainqueur = Astérix
```

Le fichier log recevra les calculs intermédiaires pour vérification : les matrices, le graphe, les calculs de chaque méthode. (le format est libre).

## 4.4 Liste ordonnée

La méthode minimax va utiliser une liste statique circulaire combinant les possibilités d'une pile et d'une file. On peut, entre autre, ajouter ou supprimer un élément en tête ou en queue. Une implémentation vous est proposée dans `file.c` et `file.h`.

Vous allez gérer deux fichiers `.c` : `element_liste.c` et `liste.c` (et les `.h` qui vont bien avec et qui sont donnés). En guise de spécification vous utiliserez les fichiers header fournis sur la page Moodle. Le code C implémente les sous-programmes correspondant aux prototypes. Les fichiers *header* ne devront pas être modifiés. Le type *Element* est défini à partie du type `t_arc_p` lui-même définit dans *global.h*. Le programme principal fourni servira de test pour approuver ce code.



La liste d'arcs sera utilisée dans les méthodes minimax et classement des paires où il faudra chercher les circuits. Un code est fourni "circuit.c", donné aussi sous forme de librairie "libcircuit.a". Dans tous les cas, vous devrez ajouter "bool circuits(liste larcs,int nbCandidats);" et "#include "pile.h" dans votre header.

*larcs* est une liste d'arcs représentant le graphe, *nbCandidats* est le nombre de sommets du graphe. Ce nombre doit correspondre aux nombre de candidats du scrutin. La fonction renvoie *true* si le graphe contient un circuit, *false* sinon.

Pour compiler avec la librairie "libcircuit.a" : "gcc listeDeVosFichiers.c -L. -lcircuit"

## 4.5 Liste des modules à programmer

- fichier `utils_sd` : allocation dynamique de tableaux, affichage de ces tableaux avec `logfp`, sous-programme utilitaires de type chercher le min, le max etc.,
- module `liste` : `element_liste.c`, `liste.c` et la gestion d'une liste
- module `uninominales` : les deux méthodes uninominales et leurs sous-programmes
- module `condorcet_minimax` : cherche un vainqueur de Condorcet, applique minimax si besoin
- module `condorcet_paires` : cherche un vainqueur de Condorcet, applique le classement des paires si besoin
- module `condorcet_schulze` : cherche un vainqueur de Condorcet, applique la méthode de Schulze si besoin
- module `jugement_majoritaire` : applique la méthode du jugement majoritaire
- module `lecture_csv` : lecture des fichiers, initialisation des structures de données
- module `main` : récupération des arguments de la ligne de commande, analyse des balises
- programme indépendant *verify\_my\_vote* : cherche dans le fichier de votes, le vote correspondant au hash d'un votant avec sa clef secrète.
- fichier `Makefile`

A ce découpage on peut éventuellement ajouter un module `utils_scrutins` qui regroupera des sous-programmes communs aux différentes méthodes de scrutin (optionnel).

## 4.6 Evaluation des modules

A chaque module est associé un nombre maximal de points à gagner. On rappelle ici (cf. document de gestion de projet) que pour obtenir le maximum des points d'un module il faut que :

1. Le code respecte les spécifications et fonctionne sans bug,
2. Le code respecte un minimum les principes du *clean code* et est commenté d'une manière utile (informatrice et concise), et possède une entête Doxygen

Le bonus est à la discrétion du tuteur. Il récompense un "plus" comme l'utilisation de git, des shell de test, un makefile poussé, etc.

### 4.6.1 Les tests (rappels)

Il est conseillé de tester les sous-programmes dès que possible (tests unitaires), et de ne les assembler entre eux que quand ils sont raisonnablement débogués. Après assemblage il faut tester ce nouveau code (tests d'intégration) et ainsi de suite avec les assemblages d'assemblages.

Des shells de test sont fournis pour vous aider à valider les codes des méthodes.

### 4.6.2 Documentation

Plutôt que de rédiger des doc papier nous allons utiliser Doxygen qui permet de compiler les commentaires du code, du moins ceux qui sont précédés des balises reconnues par le soft.

Il y a plusieurs façons de gérer ces commentaires. La plus simple est de commencer le commentaire par trois "slash" "///" suivis de la balise elle-même précédée d'un anti-slash.

Les balises les plus utiles sont :

```
\author l'auteur
\date la date
\struct to document a C struct.
\union to document a union.
\enum to document an enumeration type.
\fn to document a function.
\var to document a variable or typedef or enum value.
\def to document a define.
\typedef to document a type definition.
\file to document a file.
\brief pour décrire ce que fait le programme
\param[in] to document an incoming parameter
\param[out] to document an outgoing parameter
\param[in,out] to document an in-out parameter
\return to document the return value of a function
\pre preconditions
\post postconditions
\invariant invariants
\note to print "note" before the comment
\remarks to print "remarks" before the comment
```

**Exemple :**

```
/// \brief Fonction factorielle
/// \author Gérard Menvuuncodesibo
/// \date 9 octobre 2018
long fact(int n){
/// \param[in] n un entier positif ou nul
/// \return n!
if (n==0)
    return 1;
else
    return n*fact(n-1);
}
```

La doc est générée sous forme de documents *html* consultables avec un navigateur. Si les balises de l'auteur ou de la date sont à mettre seulement en début de fichier, il peut être intéressant de commenter les paramètres, les valeurs de retour, et de donner une indication sur le rôle des sous-programmes. Sans abus là aussi.

## 5 Annexes

### 5.1 Format CSV

CSV signifie "comma separated values". Ce format permet d'écrire un ensemble de données de type feuille de calcul dans un fichier texte. La première ligne donne les entêtes de colonnes. Les lignes suivantes donnent les valeurs. Les lignes doivent avoir le même nombre de valeurs (ou au moins de virgules). Ce type de fichier est reconnu par les tableaux qui construisent la feuille de calcul correspondante.

Les valeurs peuvent être séparées par autre chose qu'une virgule (tab, point virgule, espace, etc).

Les fichiers au format csv ne sont que des fichiers texte. Ils peuvent être lus, créés, modifiés sous éditeur.

### 5.2 Le programme de simulation de votes

Ce programme est donné. Il est programmé en Python et il permet de générer des jeux de données en simulant un vote de classement des candidats. Il génère un fichier csv du même type que celui donné par Moodle où figure les ballots. Il prend en argument un fichier texte où figurent les noms des candidats, le nombre de votes souhaités, et donne en sortie la simulation sous forme d'un fichier texte (csv) où le séparateur est un tab.

#### Utilisation :

```
python votation.py -i candidats.txt -v nombre_de_votants -o fichier.csv
```

où le fichier *candidats.txt* est un fichier texte donnant la liste des candidats (un par ligne).

Le programme votation2.py est le même mais exclus la possibilité d'avoir des ex aequo.

### 5.3 Graphes orientés et graphe de tournoi

Rappel du cours de maths discrètes de L1 :

Un graphe orienté  $G = (V, A)$  est un couple formé de  $V$  un ensemble de sommets et  $A$  un ensemble d'arcs, chaque arc étant un couple de sommets ( $A \subset V \times V$ ).

**Définitions :** Étant donné un arc  $(x, y)$ , on dit que  $x$  est l'origine (ou la source) de  $(x, y)$  et que  $y$  est l'extrémité (ou la cible) de  $(x, y)$ . On peut le représenter par une flèche allant de  $x$  vers  $y$ .

Le demi-degré extérieur (degré sortant) d'un nœud, noté  $d^+(x)$ , est le nombre d'arcs ayant ce nœud pour origine.

Le demi-degré intérieur (degré entrant) d'un nœud, noté  $d^-(x)$ , est le nombre d'arcs ayant ce nœud pour extrémité.

Autres définitions utiles :

- Chaque arc ayant une seule origine et une seule extrémité, le graphe comporte autant de degrés sortants que de degrés entrants :  $\sum_{x \in V} d^+(x) = \sum_{x \in V} d^-(x)$ .
- $x_1x_2, x_2x_3, \dots, x_{n-1}x_n$  est un chemin si et seulement si  $\forall p \in \{1, 2, \dots, n-1\}, (x_p, x_{p+1})$  est un arc.
- Le chemin  $x_1x_2, x_2x_3, \dots, x_{n-1}x_n$  est un circuit si et seulement si  $(x_n, x_1)$  est un arc.
- Un graphe complet orienté antisymétrique est un tournoi :  $\forall x, y \in V, (x, y) \in A$  où  $(y, x) \in A$
- On peut définir des graphes orientés étiquetés comme un graphe orienté et une fonction  $f : A \rightarrow N$  qui a chaque arc associe une nombre entier. Ce nombre est le poids de l'arc.

Le moyen le plus simple de représenter un graphe orienté dans le cadre de ce cahier des charges, est de faire une liste de ses arcs. Pour les méthodes minimax, paires et Schulze, les calculs pourront être faits directement sur le graphe, c'est à dire en analysant la liste des arcs.