

# RAPPORT DE PROJET

Salim GHBABRA

## Répartition du travail :

J'ai réalisé l'intégralité du projet, ayant décidé de le faire seul.

## Documentation des fonctions :

Forme du graphe : (sommet \* sommet list) list

**nbsommets** : Équivalent de `List.length`

**nbarcs** : Renvoie la somme des sommets des listes d'adjacences (seconds éléments des couples)

**ensSom** : Renvoie une liste comprenant l'ensemble des sommets (premiers éléments des couples)

**affEnsSom** : Prend en paramètre une liste de sommets et renvoie un string des noms de ceux-ci

**affEnsUtil** : Équivalent de `affEnsSom` pour une liste d'utilisateurs

**getNom** : Renvoie le nom du sommet passé en paramètre

**insertParNom, triParNom** : Fonctions nécessaires au tri par insertion (en fonction du nom)

**ensSomNom** : Renvoie une liste de sommets triés par nom en faisant appel aux fonctions de tri par insertion

**getDegSor** : Prend en paramètre un couple (sommet \* sommet list) et renvoie le nombre de sommets dans la liste d'adjacence

**insertParDeg, triParDeg** : Fonctions nécessaires au tri par insertion (en fonction du degré sortant)

**ensSomDeg** : Renvoie une liste de sommets triés par degré sortant en faisant appel aux fonctions de tri par insertion

**ensArcs** : Prend en paramètre le graphe complet et renvoie une liste de couples de sommets : (sommet\*sommet) list

**affEnsArcs** : Prend en paramètre une liste de couples de sommets et renvoie un string de la forme  
"(U1,P5)  
(U1,U8)  
(U2,P6)  
..."

**exception Introuvable** : Levée lorsque le sommet ou l'arc recherché n'est pas présent dans le graphe

**supprSom** : Renvoie la concaténation de la liste précédant et de la liste suivant le sommet supprimé

**ajoutSom** : Renvoie la concaténation d'un couple (sommet \* [ ]) au graphe

**supprOcc** : Supprime un sommet donné en paramètre d'une liste de sommets, de la même manière que la fonction supprSom

**supprArc** : Supprime un arc en faisant un appel à supprOcc sur la liste d'adjacence du sommet d'origine, ce qui entraîne la suppression du sommet d'arrivée dans la liste d'adjacence

**ajoutArc** : Recherche le sommet d'origine de l'arc dans le graphe et ajoute le sommet d'arrivée à sa liste d'adjacence

**infosom** : Cherche un sommet ayant le même nom que celui passé en paramètre et le renvoie

**affUtil** : Affiche les attributs d'un utilisateur

**affPage** : Affiche les attributs d'une page

**nbUtil** : Renvoie le nombre d'utilisateurs du graphe

**nbPage** : Renvoie le nombre de pages du graphe

**nbUP** : Renvoie un string du nombre d'utilisateurs et de pages du graphe

**ageMoyenUtil** : Renvoie la moyenne des âges des utilisateurs en divisant la somme des âges par le nombre d'utilisateurs

**appartient** : Renvoie true si le sommet passé en paramètre appartient au graphe, et false sinon

**adminsPage** : La fonction fusion fusionne deux listes d'utilisateurs en évitant les duplications grâce à la fonction appartient. La fonction aux fusionne toutes les listes d'administrateurs de pages (p.admins) du graphe et renvoie une liste d'utilisateurs uniques

**toString** : La fonction aux2 renvoie un string de la liste d'adjacence (sommet list) passée en paramètre. La fonction aux renvoie un string du graphe complet représenté sous la forme d'une liste de couple (sommet\*sommet list), en utilisant aux2 pour (sommet list). On affiche le résultat de la fonction aux grâce à un print\_string

**creaUtil** : Permet de créer un utilisateur

**crealistadmins** : Permet de créer une liste d'utilisateurs afin de créer une page. Sur la ligne

**let** Util adm = infosom l a

Le Util permet de ne traiter que les sommets de type Utilisateur

**creaPage** : Permet de créer une page à l'aide de la fonction crealistadmins

**creasommet** : Aide à la création d'un sommet en appelant soit creaUtil soit creaPage

**crealisteadj** : Permet de créer une liste d'adjacence et la renvoie sous forme de (sommet list)

**lireGraphe** : Aide à la création d'un graphe complet, en faisant appel aux fonction creasommet et crealisteadj

**top\_level** : Propose les différents services dans une interface facilitant leur utilisation

**main** : Lance la fonction top\_level avec un graphe vide en paramètre

## Problèmes rencontrés:

### ✗ Gestion d'exceptions:

Lorsque je code un bloc try with de cette forme :

```
begin try
  (top_level (supprSom l (infosom l str)))
with | Introuvable -> print_string ("Sommet introuvable!"); top_level l
end
```

Emacs l'accepte, mais lorsque je le code sous cette forme :

```
begin try
  (let k = infosom l str)
with | Introuvable -> print_string ("Sommet introuvable!"); top_level l
end
```

Emacs me renvoie une erreur de type Syntax error, pointant sur cette partie du code :

```
(let k = infosom l str) with
```

N'ayant pas trouvé de solution à ce problème, je ne peux donc pas gérer l'exception

Introuvable renvoyée par la fonction infosom.

## Travail réalisé :

L'ensemble des 13 questions demandées ont été codées, ainsi que l'interface d'utilisation mais je n'ai pas trouvé le temps et la motivation pour le PageRank (implémentation et formule).