Afin d’implémenter l’algorithme PageRank et de le tester , nous avons implémenter plusieurs fonctions telles que :

* **La fonction de normalisation** qui est une fonction qui calcule la norme euclidienne d'un vecteur. Elle prend en argument un vecteur dont on veut calculer la norme et retourne un réel qui représente la norme euclidienne du vecteur.
* Nous avons également une **fonction import\_tsv** qui importe le fichier tsv et le transforme en liste de listes. Elle retourne une Liste de listes contenant les données du fichier tsv.
* La **fonction points\_et\_transitions** qui renvoie la liste des points et la liste des transitions.
* La **fonction adjacence\_graph** qui renvoie la matrice d'adjacence du graphe.
* La **fonction PageRank** permettant de calculer les valeurs PageRank d'une matrice d'adjacence. Elle prend en argument une matrice d'adjacence du graphe , un dumping factor avec une valeur à défaut égale à 0.85, un nombre maximal d'itération avec une valeur à défaut égale à 100 et une tolérance avec une valeur à défaut égale à 1e-6. La fonction retourne une matrice des valeurs de PageRank. On commence par calculer la matrice de transition (stochastique), ensuite appliquer le facteur d'amortissement d (dumping factor), initialiser le vecteur PageRank et enfin appliquer la méthode de la puissance.
* Enfin, nous avons la **fonction personalized\_pagerank** qui permet de calculer les valeurs PageRank d'une matrice d'adjacence à partir d'un vecteur de personnalisation. Elle prend en argument une matrice d'adjacence du graphe , un vecteur de personnalisation, un dumping factor avec une valeur à défaut égale à 0.85, un nombre maximal d'itération avec une valeur à défaut égale à 100 et une tolérance avec une valeur à défaut égale à 1e-6. Dans cette fonction, on commence par vérifier si la somme de v est égale à 1, calculer la matrice de transition (stochastique) , appliquer le facteur d'amortissement d et le vecteur de préférence personnel v, initialiser le vecteur PageRank et enfin appliquer la méthode de la puissance.

**Description du programme principal :**

Dans ce programme, nous commençons par récupérer la matrice d’adjacence du graphe pour pouvoir calculer les valeurs du PageRank. On affiche également la somme des valeurs du PageRank. Ensuite, nous passons à une variation des paramètres avec une valeur de dumping factor égale à 0.15 et une tolérance égale à 0.01. On refait varier la valeur de la tolérance à une valeur plus petite (tol = 0.0001). On fait aussi varier le dumping factor à 0.85 avec les valeurs de tolérance décrites ci-dessus.

**Interprétations des résultats:**

On remarque que la page ayant le score de PageRank le plus élevé est la page "United States" avec un score de PageRank presque 2 fois plus élevé que la seconde page qui est la page "Europe".

On peut donc en déduire que la Page United States est la page la plus visité par les participants du jeu. La page United States est donc une page que les joueurs estiment comme étant une page très intéressante pour obtenir d'autres liens leur permettant d'approcher leur but.

D'ailleurs le top 5 ordonné des pages les plus visités dans les parcours des joueurs est le suivant:

1- United\_States

2- Europe

3- United\_Kingdom

4- England

5- World\_War\_II

En ce qui concerne la variation des paramètres, avec une tolérance trop grande et un dumping factor trop faible, on obtient un score de PageRank identique pour toutes les pages.

Plus la tolérance est faible et plus les scores des premières pages sont grands et ceux des dernières pages sont petits. Une tolérance très faible permet d'obtenir de meilleurs résultats plus précis.

Plus le dumping factor est élevé et plus la différence de score entre les premières et dernières pages (en terme de classement par score de PageRank) augmente.

Si le dumping factor est trop faible, toutes les pages ont un score de PageRank proche, tandis qu'a l'inverse on observe une réellel différence.

On a regardé l'impact en terme de temps d'exécution du changement des paramètres, mais avec les paramètres que l'on a pris nous n'avons pas observé de différence significative. En revanche, il est logique de penser que plus la tolérance est petite et plus le temps d'exécution est long.

**Personnalisation:**

Par la suite, on a fait une personnalisation des premiers nœuds et des derniers nœuds du graphe et en plus un calcul de PageRank.

Lorsqu’on personnalise les nœuds avec un score PageRank élevé, ces nœuds sont considérés comme les plus importants ou les plus pertinents dans le réseau. Personnaliser ces nœuds peut avoir un impact significatif sur la qualité des résultats de recherche. On va donc ici personnaliser le top 5 des nœuds en terme de score PageRank.

Par rapport au PageRank original, les pages que l'on a décidé de personnaliser et les pages connectées à ces dernières ont vu leur score de PageRank augmenter. De ce fait l'ordre des pages a changé. Les premières pages n'ont pas bougé étant donné que ce sont elles que nous avons décidé de personnaliser. En revanche, pour les autres pages, il y a eu un changement d'ordre.

Nous avons aussi décidé de personnaliser les dernières pages afin de voir les conséquences que cela aura sur les score de PageRank. Cela a eu pour effet de totalement changer les scores et le classement des pages. En effet, le top 5 est totalement différent. Ce sont les pages personnalisées qui se retrouvent maintenant en tête de classement de score PageRank.

Pour finir, nous avons également fait des tests pour vérifier si nos résultats sont bons. Pour cela, nous avons utilisé la library PageRank sur python (from fast\_pagerank import pagerank\_power) et nous avons les mêmes résultats que les tests réalisés avec les fonctions que nous avons implémenté.