Isack Breton BREI

Antoine Labrie LABA2123

GIF270 – APP5: Structures de données complexité

Nous avons utilisé plusieurs structures de données pour la réalisation de cet APP. Premièrement, pour la création de notre N-gramme, nous avons utilisé un tableau de hachage. Nous avons donc entré chaque mot des textes analysés dans le tableau en tant que clé et ceux si sont associés à un index pour chaque mot unique. Chaque clé du tableau avait également une valeur associée qui correspond au nombre de fois que le mot était présent dans le texte. Le même principe s’appliquait pour des N-gramme où N est plus grand qu’un, seulement dans ce cas, chaque clé correspond à une séquence de N mots. Dans le cas de cet APP, le tableau de hachage était la structure de donnée la plus simple à utiliser pour la création d’un N-gramme. Ça nous permet de stocker rapidement les données et d’aller les chercher avec un maximum de complexité linéaire O(n).

Pour la génération de texte aléatoire, la structure de donnée que nous avons utilisée était le graphe. Celui-ci permettait de stocker nos données afin de produire une chaine de Markov. Celle-ci permettait ensuite de facilement générer un texte en suivant la chaine de Markov pour déterminer quel mot devrait être inséré dans notre texte aléatoire.

La complexité de l’algorithme pour le N-gramme était linéaire, O(n) en notation grand O, car le tableau de hachage était produit avec une boucle passant par tous les mots des textes une seule fois. La complexité était la même lorsque l’on veut aller chercher une donnée dans ce tableau. Le maximum de complexité est O(n) et le minimum est O(1) dans le cas où le mot cherché est le premier vérifié dans le tableau.

Pour l’algorithme du ne mot le plus fréquent et de la ne séquence de deux mots la plus fréquente, la complexité est la même. Premièrement, la fonction sort() de python est utilisé pour classer le N-gramme en ordre du plus fréquent au moins fréquent et cette fonction a une complexité de O(n log n). Ensuite, le tableau classé est parcouru jusqu’au ne caractère, donc la complexité est de O(n) au maximum et de O(1) au minimum.

Le tableau suivant montre nos résultats pour le mot et le digramme les plus utilisés par chaque auteur d’après l’analyse effectuée avec notre programme.

|  |  |  |
| --- | --- | --- |
| Auteur | Mot le plus utilisé | Digramme le plus utilisé |
| Balzac | x | X |
| Hugo | X | X |
| Jean | X | X |
| Ségur | x | X |
| Verne | X | X |
| Voltaire | X | X |
| Zola | X | x |