Maret Luka, Naudts Angelo

Rapport choix algorithme

Algorithme de tri :

Choix de l’algorithme : Tri fusion

La fonction merge permet de réunir de manière triée les 2 listes chaînés. Il prend en paramètre les 2 listes chainés, et va les trier et les regrouper. C’est le même fonctionnement qu’avec les tableaux.

Nous avons choisi cet algorithme car c’est le plus rapide. Une image contenant texte

Description générée automatiquement

Complexité :

Fonction merge, en fonction de n=len(start)+len(end)

Temps O(n), espace O(n)

Une image contenant texte

Description générée automatiquement

Complexité :

Fonction merge\_sort, en fonction de n le nombre d’éléments de la liste chaînée.

si T(n) désigne le nombre d’appels à la fonction merge\_sort par une liste chaînée de taille n : T(n) <= log2(n)

(nombre de fois que l’on peut diviser n par 2)

Temps total de tri\_fusion : O(nlog2(n))

Algorithme du search :

Nous avons choisi d’utiliser l’algorithme de recherche séquentielle car notre liste d’origine n’est pas triée. On parcours la liste chaînée jusqu’à ce qu’on tombe sur x.

Une image contenant texte

Description générée automatiquement

Complexité :

Pire des cas : si x n’est pas dans list

Temps : O(n) (n taille de la liste\_chaînée)

Espace : O(n)

Algorithme du filter :

Nous avons aussi choisi l’algorithme de recherche séquentielle cependant, son fonctionnement n’est pas exactement comme celui du search. Pour cette algorithme, nous utilisons les régex et nous créons une nouvelle liste chaînée avec les éléments filtrés. Une image contenant texte

Description générée automatiquement

Complexité :

Complexité :

Pire des cas : si regex n’est pas dans list

Temps : O(n) (n taille de la liste\_chaînée)

Espace : O(n)