

Naoufal Acharki

Nicolas Lagaillardie

Compte rendu TP

Ce programme sous Python permet de générer une table de hashage de taille $701 * 509$ puis d'analyser les entrées utilisateurs pour détecter les mots dans ou hors dictionnaire. Il agit en 4 grandes étapes : récupération des mots du fichier dictionnaire, extraction de l'alphabet, placement des mots du dictionnaire dans la table et analyse des entrées utilisateur. Les trois premières parties tirent profit du multi-threading afin d'accélérer en grande partie les traitements.

Après avoir testé une fonction de hashage parfait (perfect hash function), nous avons décidé de nous orienter vers une fonction de hashage linéaire, puis du double hashage afin de limiter au mieux les collisions. Nous arrivons au pire à 1350 collisions, ce qui réduit 174 fois au moins la taille de la liste à analyser. Si la première fonction a été abandonnée, c'est parce qu'il fallait trouver des « nombres qui allaient bien », mais même avec une recherche du minimum par la méthode de Monte-Carlo, nous arrivions à un maximum de collisions à environ 2500, ce qui était trop élevé.