Rapport de Projet

Algorithme d’Huffman

Rémi Couzi & Léo Portet

Algorithme de Huffman explications ? Réalité ou Illusion ?

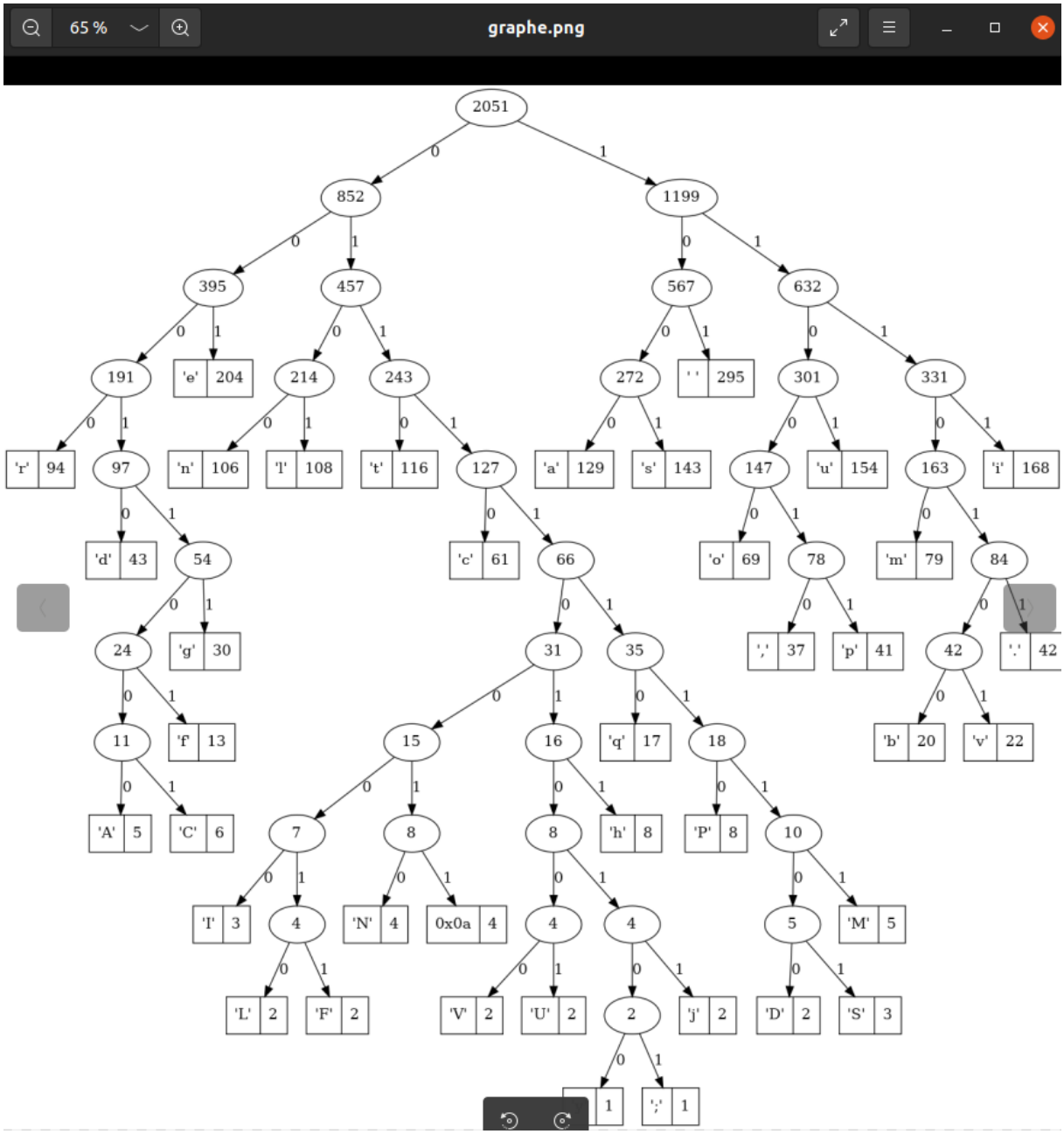
L’algorithme de Huffman est un des plus connu concernant la compression / décompression, son fonctionnement est le suivant : On parcourt le texte à compresser et on récupère les occurrences de chaque caractère. Vient maintenant l’étape la plus importante, la création de l’arbre de Huffman, il suffit de récupérer les 2 caractères les moins présents (occurrence la plus faible), on crée un arbre de Huffman les contenant et on l’insère (en tri) dans notre liste, et ce, jusqu’à ce qu’il n’y reste plus qu’un élément, qui est l’arbre final de Huffman. Une fois cet arbre obtenu, il est aisé de récupérer le code binaire de chaque caractère. Enfin, il ne reste plus qu’à remplacer ce caractère par son code dans le fichier compressé. Pour décompresser la méthode est similaire, la différence est que l’occurrence de chaque caractère est donnée explicitement dans l’en-tête du fichier compressé.

Niveau code, nous avons procédé comme suis :

Nous avons mis une interface pour que l'utilisateur choisisse s'il veut compresser ou décompresser un fichier, puis le nom du fichier.

Compression :

Nous avons ensuite mis une fonction permettant de lire le fichier fourni par l'utilisateur et de remplir un tableau statique de dimension 128 auquel chaque case correspond à un caractère, une occurrence, et une expression binaire (chaîne de caractère) que l'on complétera plus tard une fois l'arbre de Huffman créé. Une fois le tableau rempli, on crée un nœud d'ABR pour chaque case, et on l'insère dans cet arbre, trié. Une fois ce 1er arbre complet, il nous suffit de récupérer ses 2 plus petits nœuds (les plus à gauche, étant donné qu'il est trié), on crée un Nœud de Huffman ayant pour fils chacun des deux qu'on transforme en Huffman.



Une fois notre arbre de Huffman construit, il suffit de le parcourir pour chaque caractère pour récupérer le code en binaire (0 vers la gauche, 1 vers la droite), une fois le code obtenu, on le met dans le tableau initial, pour chaque caractère. Nous pouvons maintenant commencer à écrire dans l'archive, en mettant d'abord le nombre de caractères différents encodés (nombre obtenu grâce aux occurrences dans le tableau), puis sur chacune des lignes suivantes chaque caractère du tableau ainsi que son occurrence. Il ne suffit plus maintenant qu'à reparcourir le texte à compresser et d'écrire le code binaire correspondant à chaque caractère dans notre nouveau fichier.

Décompression :

Pour la décompression, nous parcourons d'abord le fichier compressé pour récupérer le nombre de caractère différents ainsi que leur occurrence (dans l'en-tête du fichier compressé), on complète le tableau comme dit précédemment avec le caractère et son occurrence, ensuite la méthode est identique, on crée l'ABR, puis l'arbre de Huffman, on récupérer les codes binaires des caractères que l'on met dans le tableau. Une fois le tableau obtenu, on continue de parcourir le fichier compressé et à l'aide d'une fonction récursive qui associe chaque 0/1 à un déplacement dans l'arbre nous récupérons les caractères à chaque fois que le chemin se termine, on écrit ce caractère dans notre fichier décompressé, et ce, jusqu'à ce que l'archive soit lu entièrement.