TEHARI Ayoub
L3 informatique – UGA

Compte rendu de l’apnée 3 et 4
Algorithme de Huffman

Introduction :

Ce compte rendu résume ce que est utilisé comme fichiers de tests, et les problèmes rencontrés lors de la réalisation de l’algorithme de Huffman, vu en cours.

Problèmes rencontrés :

Utilisation de la fonction ArbreVide au lieu de la fonction EstVide lors de la phase de construction des codes.

Ça marche pas avec les fichiers générés aléatoirement, ni avec des fichiers comme les .pdf

Les fichiers tests :

script.sh :

```
#!/bin/bash


#for i in $(seq 1 100)
#do
#       name=test_$i.txt
#       ./aleatoire > $name
#done


for file in *.txt *.pdf
do
        file1=$(basename $file .pdf)
        encode=$(basename $file1 .txt).huff
        echo $file
        decode=$(basename $file .txt).res
        ./huff_encode $file $encode 1>/dev/null
        ./huff_decode $encode > $decode
        if diff $decode $file
        then
                echo 'OK'
                rm $encode $decode
        else
                echo "KO"
        fi
done
```

```
Le principe ici est de faire le codage ensuite le décodage pour au final
arriver au même résultat que l’original.
```

10000a.c :

```
#include <stdio.h>

int main () {
        int i;
        for (i=0 ; i<10000 ; i++) {
                printf("a");
        }
```

```
	return 0;
}
```

Ce test ne réussit pas car on a besoin d'au moins de 2 caractères pour pouvoir les distinguer et faire l'arbre.

aleatoire.c :

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>


int main () {
	srand(getpid());
	int limits[255];
	int i, j, l, v=0, N= 25600;
	for (i=0; i<255; i++) {
		l = (N - v ) - (255 - i);
		limits[i] = rand() % l + v;
		v = limits[i];
	}
	int res;
	for (i = 0 ; i<10000000 ; i++ ) {
		res = rand () % 25600;
		j=0;
		while (j<255 && limits[j] < res) {
			j++;
		}
		printf("%c", j);
	}
	return 0;
}
```

Je n'ai pas réussit à détecter où est l'erreur pour que le fichiers décodé et l'original soient identiques

Structure de l'arbre de Huffman :

Nœuds internes : ont l'étiquette 0 .
Feuils : ont le caractère correspondant comme étiquette.

Conclusion :

J'ai remarqué sur les tests qui marchent que la taille de ces fichiers est réduite à presque la moitié quand on applique l'algorithme de Huffman dessus.