

Corrigé série 7

1 Exercice 1

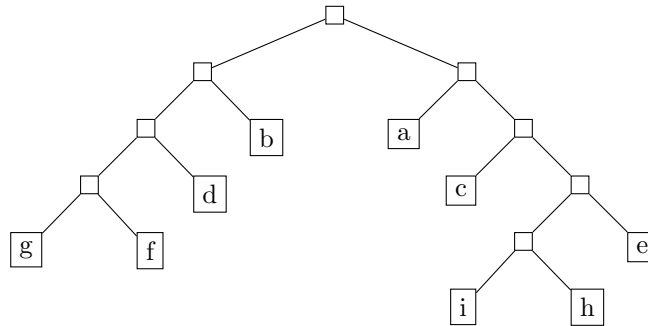
1.1 Question A

La longueur minimale est de 4. Encodage possible des caractères :

a	b	c	d	e	f	h	i
0001	0010	0011	0100	0101	0110	0111	1000

1.2 Question B

Arbre :



En posant que les branches allant vers la gauche vaut 0 et ceux de droite 1, on a l'encodage suivant pour les caractères :

1. a : 10
2. b : 01
3. c : 110
4. d : 001
5. e : 1111
6. f : 0001
7. g : 0000

8. h : 11101

9. i : 11100

La taille moyenne de l'encodage est le suivant :

$$\begin{aligned}\mu &= \frac{1}{100} \sum_{i=1}^9 l_i f_i \\ &= \frac{1}{100} (2 \times 26 + 2 \times 24 + 3 \times 13 + 3 \times 12 + 4 \times 7 + 4 \times 6 + 4 \times 5 + 5 \times 4 + 5 \times 3) \\ &= \frac{282}{100} = 2,82\end{aligned}$$

Donc, la taille moyenne de l'encodage est de 2,82.

2 Exercice 2

Tâche triée en fonction des pénalités :

Tâche	1	2	5	4	3	6	7	8
Échéance	3	2	4	3	2	5	1	7
Pénalité	95	85	75	60	55	45	40	30

Insertion tâche #1 :

i			1					
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	0	1	1	1	1	1	1

Insertion tâche #2 :

i		2	1					
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	2	2	2	2	2	2

Insertion tâche #5 :

i		2	1	5				
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	2	3	3	3	3	3

Insertion tâche #4 :

i		2	4	1	5			
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	3	4	4	4	4	4

Insertion tâche #3 :

i		3	4					
		2	1	5				
d_i	1	2	3	4	5	6	7	8
$N_i(F)$	0	1	3	4	4	4	4	4
$N_i(F)$	0	2	4	5	5	5	5	5

Insertion tâche #6 :

i		3	4					
		2	1	5	6			
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	3	4	5	5	5	

Insertion tâche #7 :

i		3	4					
	7	2	1	5	6			
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	3	4	4	4	4	
$N_i(F)$	1	2	4	5	5	5	5	

Insertion tâche #8 :

i		3	4					
	7	2	1	5	6		8	
d_i	1	2	3	4	5	6	7	
$N_i(F)$	0	1	3	4	5	5	6	

Ordre des tâches à exécuter :

$$t_2 \rightarrow t_1 \rightarrow t_4 \rightarrow t_5 \rightarrow t_6 \rightarrow t_8$$

Coût pénalité : $t_7 + t_3 = 55 + 40 = 95$.

3 Exercise 3

Variante avec dictionnaire :

1. **Fonction** TROUVEROCCURENCEPALINDROME (L)
2. $D \leftarrow \text{CreeDictionnaire}()$
3. $n \leftarrow \text{Taille}(L)$
4. Pour $i \leftarrow 1$ haut n faire
5. Si Contient(D, L[i]) alors
6. $D[L[i]] \leftarrow D[L[i]] + 1$
7. Sinon
8. $D[L[i]] \leftarrow 1$
9. Fin Si
10. Fin pour
11. $total \leftarrow 0$
12. $max \leftarrow 0$
13. Pour $c \in D.cles()$ faire
14. Si $D[c] \bmod 2 = 0$ alors
15. $total \leftarrow total + D[c]$
16. Sinon si $max < D[c]$ alors
17. $max \leftarrow D[c]$
18. Sinon
19. $total \leftarrow total + D[c] - 1$
20. Fin Si
21. Fin Pour
22. Renvoyer $total + max$
23. Fin Fonction

Variante avec table de hachage :

1. **Fonction** TROUVEROCCURENCEPALINDROME (L)
2. $H \leftarrow \text{CreeTableHachage}(26)$
3. $n \leftarrow \text{Taille}(L)$
4. Pour $i \leftarrow 1$ haut n faire
5. Si Contient(H , $L[i]$) alors
6. $H[L[i]] \leftarrow H[L[i]] + 1$
7. Sinon
8. $H[L[i]] \leftarrow 1$
9. Fin Si
10. Fin pour
11. $total \leftarrow 0$
12. $max \leftarrow 0$
13. Pour $c \in H.cles()$ faire
14. Si $H[c] \bmod 2 = 0$ alors
15. $total \leftarrow total + H[c]$
16. Sinon si $max < H[c]$ alors
17. $max \leftarrow H[c]$
18. Sinon
19. $total \leftarrow total + H[c] - 1$
20. Fin Si
21. Fin Pour
22. Renvoyer $total + max$
23. Fin Fonction

4 Exerice 4

Approche gloutonne possible : faire un parcours en largeur, mais quand on regarde les voisins du noeuds qui sont visités, on regarde aussi sa couleur et la couleur du noeud courant est le plus petit nombre qui n'est pas utilisé pour ses voisins. Dans le cas contraire, on lui attribut une nouvelle couleur.