4.1

TD Complex 4

4.1 :

data [0...n-1] n caractères, encodés sur 8 bits (256 caractères différents)
↳ 8n bits avant compression

Hist(data, n)

H[0,255] ← 0

for i ← 0 to n-1 :

  H[data[i]] ← H[data[i]] + 1

return H

$\begin{vmatrix} \Theta(1) \\ \Theta(n) \\ \Theta(n) \\ \Theta(1) \end{vmatrix}$

4.2

$T(n) = \Theta(n)$

4.3

Comptabiliser le nombre de caractères différents dans data connaissant H.

p ← 0

for i ← 0 to 255 do

  p ← p + (H[i] > 0)

return p

$T(n) = \Theta(1)$

4.4 :

$b = \lceil \log_2(p) \rceil$ Nombre de bits necessaires pour encoder p valeur.

4.5 :

Après compression : $256 + n \times \lceil \log_2(p) \rceil$

$$\frac{256 + n \times \lceil \log_2(p) \rceil}{8n} \xrightarrow[n \to +\infty]{} \frac{\lceil \log_2(p) \rceil}{8}$$

4.2

$\underbrace{1101}_{e}\underbrace{1100}_{f}\underbrace{1100}_{f}\underbrace{0}_{a}\underbrace{100}_{c}\underbrace{1101}_{e}$

Code prefixe de p char ≡ arbre binaire complet de p feuilles.

3) Nombre de Catalan : $B_n = \frac{1}{n+1}\binom{2n}{n}$: Nombres d'arbres binaire complet avec n+1 feuilles.

$p = B_{p-1} \times p! = \frac{1}{p}\binom{2(p-1)}{p-1} \times p!$

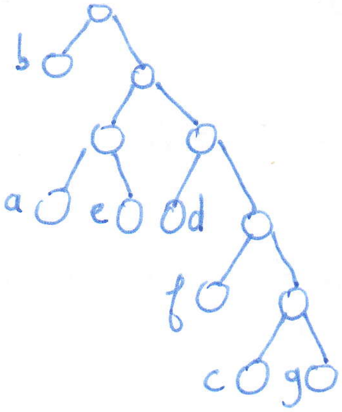
$B_n \underset{n+\infty}{\sim} \frac{4^n}{n\sqrt{n\pi}}$ | $n! \underset{n\to+\infty}{\sim} \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$

↳ $\left(\frac{4p}{e}\right)^{p-1}$

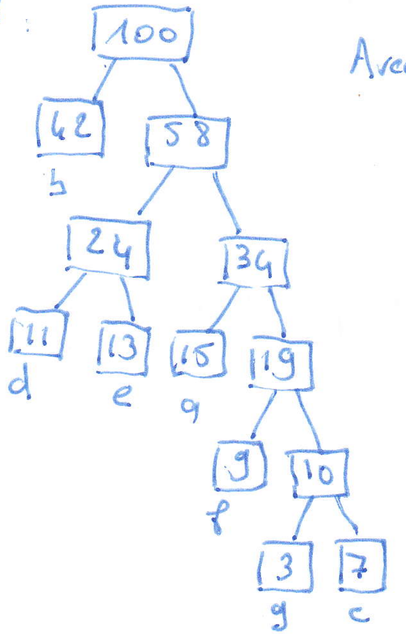
4) p caractères → arbre binaire complet a une hauteur max de (p-1).
↳ codes d'au plus (n-1) bits

Si un caractères a un support de ≥ p bits, l'arbre n'est plus complet, donc le code n'est plus optimal.

5)



4.3 :



Avec un tas : $O(p\log(p))$