PARTIE 2 - Recherche dans un dictionnaire

2.A.

```
#type info = {cle : int vect ; valeur : int}
and noeudinterne = Nil | Feuille of info | Noeud of (noeudinterne vect)
and karbre = noeudinterne ; ;

Le type info est défini.
Le type noeudinterne est défini.
Le type karbre est défini.
```

2.A.1) Pour rechercher une clé, il faut parcourir l noeuds (où l représente la longueur commune des clés) et arriver à la feuille. Soit au total $l+1$, c'est à dire la hauteur de l'arbre.

2.A.2) S'il y a n mots de longueur l sur un alphabet à k symboles, la hauteur de l'arbre est égale à l, chaque noeud interne est un tableau de longueur k. Il y en a au plus 1 à la racine, k au niveau 1, k^2 au niveau 2 etc...
Si $n = k^l$ (nombre maximum de clés de longueur l sur cet alphabet), il y aura $\dfrac{k^{l+1}-1}{k-1}$ tableaux de longueur k (les noeuds internes) et $n = k^l$ mots de longueur l (les feuilles).

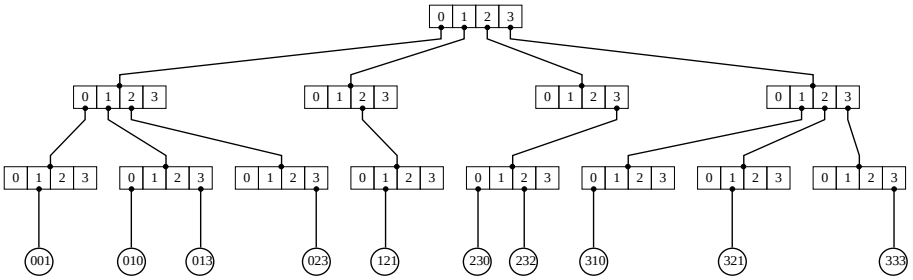
2.B.
2.B.1) Type arbre comprimé :

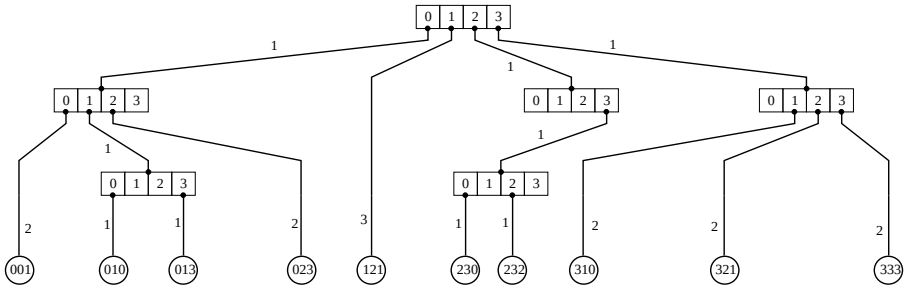
```
#type nouveaunoeud = Nil | Feuille of int*info | Noeud of int*(nouveaunoeud vect)
and karbre_comprime=nouveaunoeud ; ;

Le type nouveaunoeud est défini.
Le type karbre_comprime est défini.
```

Recherche dans un arbre comprimé :

```
let rec recherche mot dictionnaire =
match dictionnaire with
  | Nil -> false
  | Feuille (x,m) when m.cle=mot -> true
  | Noeud (x,t) -> recherche (sub_vect mot 1 (vect_length mot -1)) t.(mot.(0)) ; ;
```

2.B.2)





2.B.3) Le nombre de tableaux est alors inférieur à $n-1$.
En effet, tous les noeuds internes ont au moins deux fils. Montrons que dans un tel arbre, le nombre de noeuds internes est inférieur (strictement) au nombre de feuilles.
Un arbre reduit à une feuille ne possède aucun noeud interne.
Soit A un arbre ayant n feuilles ($n>1$)et p noeuds internes, la racine a donc au moins deux fils. Soient A_1, A_2, ..., A_k ($k \geq 2$) les branches issues de la racines, n_i et p_i le nombre de feuilles et le nombre de noeuds internes de la branche A_i. Par induction structurelle, pour tout i, $p_i \leq n_i - 1$.

$$p = \sum_{i=1}^{k} p_i + 1 \qquad n = \sum_{i=1}^{k} n_i \qquad p \leq \sum_{i=1}^{k} (n_i - 1) + 1 = n - k + 1 \leq n - 1$$

2.C.
2.C.1) A chaque dictionnaire correspond un k-arbre. Le nombre de dictionnaires est $C_{k^l}^n$.
Pour que la profondeur soit strictement supérieure à d, il faut et il suffit qu'il existe deux mots du dictionnaire ayant les mêmes $d-1$ premiers symboles.
Si $n > k^{d-1}$ c'est toujours vrai donc $N_d = C_{k^l}^n$, sinon le nombre de dictionnaires pour lesquels tous les mots ont des préfixes de longueur $d-1$ distincts est $C_{k^{d-1}}^n k^{n(l-d+1)}$, donc les dictionnaires dont le k-arbre comprimé est de profondeur strictement supérieure à d est

$$N_d = C_{k^l}^n - C_{k^{d-1}}^n k^{n(l-d+1)} = C_{k^l}^n - \frac{k^l}{n!} \prod_{i=0}^{n-1} \left(1 - \frac{i}{k^{d-1}}\right)$$

Or $\dfrac{k^l}{n!} \geq C_{k^l}^n$ donc :

$$N_d \leq C_{k^l}^n \left[1 - \prod_{i=0}^{n-1} \left(1 - \frac{i}{k^{d-1}}\right)\right]$$

Cette dernière inégalité étant vrai également si $n > k^{d-1}$ car l'un des termes du produit est alors nul et le deuxième membre vaut $C_{k^l}^n = N_d$

2.C.2)

$$\overline{d_n} = \sum_{d \geq 1} d(q_{d-1} - q_d) = \sum_{d \geq 0} q_d$$

$$q_d = \frac{N_d}{C_n^{k^l}} = 1 - \prod_{i=0}^{n} \left(1 - \frac{i}{k^{d-1}}\right) \leq 1 - e^{\frac{-n^2}{k^{d-1}}}$$

Ce dernier terme est ≤ 1 et aussi $\leq \dfrac{n^2}{k^{d-1}}$ (car, pour tout x, $e^x \geq 1 + x$). Donc :

$$q_d \leq min\left(1, \frac{n^2}{k^{d-1}}\right)$$

On en déduit donc :

$$\overline{d_n} \leq \sum_{d \geq 0} q_d \leq \sum_{d \geq 0} min\left(1, \frac{n^2}{k^{d-1}}\right)$$

Soit $\alpha = \log_k n$, si $d \geq 2\alpha + 1$ (c'est à dire $d \geq \lceil 2\alpha \rceil + 1$), $min\left(1, \frac{n^2}{k^{d-1}}\right) = \frac{n^2}{k^{d-1}}$, sinon, $min\left(1, \frac{n^2}{k^{d-1}}\right) = 1$, soit

$$\overline{d_n} \leq \lceil 2\alpha \rceil + \sum_{d \geq \lceil 2\alpha \rceil + 1} \frac{n^2}{k^{d-1}}$$

$$\sum_{d \geq \lceil 2\alpha \rceil + 1} \frac{n^2}{k^{d-1}} = \frac{n^2}{k^{\lceil 2\alpha \rceil}} \frac{k}{k-1} \leq \frac{k}{k-1} = O(1)$$

Donc,

$$\overline{d_n} \leq 2\lceil \log_k n \rceil + O(1)$$

car $\lceil 2\alpha \rceil \leq 2\lceil \alpha \rceil$.