Exercice 2. Étude d'une « suite univers »

Cet exercice a été fait par Matthieu SOLNON, merci à lui !

Question 1 –

```
def champ(N):
    """champ(N: int) -> str"""
    s = ""
    for i in range(N+1):
        s += str(i)
    return s
```

Question 2 –

```
# Utiliser la fonction champ donnerait une moins bonne complexité
def champ2(N):
    """champ2(N: int) -> str"""
    s = ""
    i = 0
    while len(s) < N+1:
        s += str(i)
        i += 1
    return s[:N+1]
```

Question 3 –

```
def cherche(mot, texte):
    """cherche(mot: str, texte: str) -> bool"""
    t = len(mot)
    for i in range(len(texte)-t+1):
        if mot == texte[i:i+t]:
            return True
    return False
```

Question 4 – Le nombre de tours dans la boucle `for` est au plus n. Si un test d'égalité entre deux chaînes de caractères s'exécute en temps constant, alors chaque tour de boucle s'exécute en temps constant. On obtient bien une complexité en $\mathcal{O}(n)$.

Question 5 –

```
def champred(N):
    C = ""
    for k in range(N+1):
        if not cherche(str(k) ,C):
            C = C + str(k)
    return C
```

Question 6 – Pour cette question, il semble qu'on ait besoin du résultat de la question 12 : soit M le nombre de chiffres dans N, alors il existe une constant α telle que $M \leqslant \alpha \ln(N)$.

La boucle `for` de la fonction `champred` fait $N+1$ tours. Dans chaque tour de boucle, la taille de `C` est au plus $(N+1) \times M \leqslant \alpha(N+1)\ln(N)$. Ainsi, $\boxed{\text{la complexité est en } \mathcal{O}(N^2 \ln(N))}$.

Question 7 – À chaque étape de la fonction `champred`, on concatène `str(k)` seulement si cette chaîne de caractères n'apparaît pas dans `C`. Ainsi, lorsqu'on teste si une séquence `s` apparaît déjà dans `C`, la longueur de `s` est inférieure ou égale à `L` (car `k` $\leqslant$ `N` et donc la longueur de `k` est inférieure ou égale à la longueur de `N`). D'où le résultat.

Question 8 –
⋆ Pour rechercher une séquence `s` dans `L`, on est contraints d'utiliser une recherche séquentielle de coût $\boxed{\text{linéaire en la taille de } \texttt{L}}$.
⋆ Pour obtenir la même information $\boxed{\text{avec le dictionnaire } \texttt{D}, \text{ le coût est constant}}$. En effet, rechercher une clé donnée dans un dictionnaire s'effectue en temps constant.

Question 9 –

```
def initdico(N):
    """initdico(N: int) -> dict[str: bool]"""
    d = {}
    for i in range(N+1):
        d[str(i)] = False
    return d
```

Question 10 –

```
def longueur(N):
    """longueur(N: int) -> int"""
    if N == 0:
        return 1
    M = 0
    while N > 0:
        M += 1
        N //= 10
    return M
```

Question 11 – On suppose $N \geqslant 2$. Comme M est défini comme le nombre de chiffres dans N, on a :

$$N \geqslant 10^{M-1}$$

Donc :

$$\ln(N) \geqslant (M-1)\ln(10)$$

D'où :

$$M \leqslant \frac{\ln(N)}{\ln(10)} + 1 = \frac{\ln(N)}{\ln(10)} + \frac{\ln(2)}{\ln(2)} \leqslant \frac{\ln(N)}{\ln(10)} + \frac{\ln(N)}{\ln(2)}$$

Ainsi, $\boxed{\alpha = \dfrac{1}{\ln(10)} + \dfrac{1}{\ln(2)}}$ convient.

Question 12 –

```
1  def champred2(N):
2      M = longueur(N)
3      # initialisations :
4      C = ''
5      D = initdico(N)
6      for k in range(N+1):
7          if D[str(k)] == False :
8              for chiffre in str(k):
9                  # ajout de chiffre dans C :
10                 C = C + chiffre
11                 for j in range(1,M+1):
12                     # déclarer la séquence des j derniers
13                     # caractères de C comme déjà vue :
14                     D[C[-j:]] = True
15     return C
```

Question 13 – Dans la fonction `champred2`, il y a trois boucles imbriquées :

- La boucle de la ligne 6 qui effectue `N+1` tours.
- La boucle de la ligne 8 qui effectue au plus $\alpha \ln(k) \leqslant \alpha \ln(N)$ tours (voir question 12)
- La boucle de la ligne 11 qui effectue au plus $M \leqslant \alpha \ln(N)$ tours (voir question 12)

Chaque ligne prise individuellement s'exécute en temps constant, donc le temps d'exécution est en $\mathcal{O}(N \ln^2(N))$.

Remarque. En réalité, l'instruction « `C[-j:]` » ne s'exécute pas en temps constant, mais l'énoncé semble supposer que c'est une opération élémentaire.

Question 14 –

```
def champred2_bis(N):
    M = longueur(N)
    C = ''
    D = initdico(N)
    X = 0
    for k in range(N+1):
        if D[str(k)] == False :
            X += 1
            for chiffre in str(k):
                C = C + chiffre
                for j in range(1,M+1):
                    D[C[-j:]] = True
    return X
```

Question 15 –

```
def comptage(chaine):
    """comptage(chaine: str) -> dict[str: int]"""
    d = {}
    for i in range(10):
        d[str(i)] = 0
    for c in chaine:
        d[c] += 1
    return d
```