Devoir n° 4 *

Claudéric DeRoy (p1174700)   Alexandre Pachot (p0774809)

2 décembre 2020

Question 1

Les choix possibles sont :

Entrée Il y a 8 choix, soit une des 7 entrées, soit rien.

Plat principal + salades Il y a trois options :

4/

- 4 plats et 9 salades, soit 36 choix ;
- 5 plats et aucune salade, soit 5 choix ;
- 6 plats et soit une des 9 salades soit rien, ce qui fait $6 \times (9 + 1) = 60$ choix.

Ce qui fait un total de $36 + 5 + 60 = 101$ choix pour le plat principal et la salade.

Desserts Il y a un choix entre 10 desserts.

Café Soit un café, soit rien. Ce qui fait 2 choix.

Le nombre de choix total est $8 \times 101 \times 10 \times 2 = 16\,160$. S'il vient tous les jours à ce restaurant. Il peut manger pendant 44 ans sans faire deux fois le même choix !

1.1

*IFT 1065 – Structures discrètes en informatique – Automne 2020 – Margarida Carvalho

4

Question 2

On a 31 bâtiments, et si l'on veut les numéroter en ordre croissant sans avoir deux numéros consécutifs, alors on a :

$$\begin{array}{ll}
\text{Bâtiment } 1 & \rightarrow \text{numéro } 1 \\
\text{Bâtiment } 2 & \rightarrow \text{numéro } 3 \\
\text{Bâtiment } 3 & \rightarrow \text{numéro } 5 \\
\quad\vdots & \qquad\quad\vdots \\
\text{Bâtiment } i & \rightarrow \text{numéro } 2i-1 \\
\quad\vdots & \qquad\quad\vdots \\
\text{Bâtiment } 30 & \rightarrow \text{numéro } 59 \\
\text{Bâtiment } 31 & \rightarrow \text{numéro } 61
\end{array}$$

Mais, nous avons seulement 60 numéros. On a donc $n > k$ où n est le nombre de maison et k est le nombre de numéros possible. Alors par principe du pigeonnier on a forcément deux numéros consécutifs.

Question 3

4

On va l'algorithme par récurrence. Pour cela, nous avons besoin d'une procédure qui permet d'extraire une chaine de caractère.

Convention :

Opérateur d'égalité : $=$
Opérateur d'affectation : $\leftarrow$
Opérateur de concaténation : $+$

Algorithme SousChaine

Résultat : Extrait une partie d'une chaine de caractères
Entrées : Une chaine de caractères, l'index du premier caractère, l'index du dernier élément
Sorties : La chaine de caractères entre les deux index

Procédure SousChaine *(chaine, debut, fin)*
  $sousChaine \leftarrow$ "" ;
  pour $index \leftarrow debut$ **à** fin **faire**
    $sousChaine \leftarrow sousChaine + chaine[index]$;
  fin
  retourner $sousChaine$;
Fin SousChaine

Algorithme InverserChaine

Résultat : Inverse les caractères d'une chaine de caractères
Entrées : Une chaine de caractères
Sorties : La même chaine, mais inversée

```
Procédure InverserChaine (chaine)
    longueur ← len(chaine) ;
    si longueur = 1 alors
        retourner chaine ;
    fin
    sinon si longueur = 2 alors
        retourner chaine[2] + chaine[1] ;
    fin
    sinon
        dernierCaractere ← chaine[longueur] ;
        premierCaractere ← chaine[1] ;
        leReste ← SousChaine(chaine, 2, longueur − 1) ;
        retourner
          dernierCaractere + InverserChaine(leReste) + premierCaractere ;
    fin
Fin InverserChaine
```

4/

Question 4

$f(n)$ est la somme des n^3 premiers termes. Il y a en tout n^3 termes.

$$f(n) = \underbrace{1 + 2 + 3 + \cdots + n^3}_{n^3 \text{ termes}}$$

$$1 + 2 + 3 + \cdots + n^3 \leq n^3 + n^3 + n^3 + \cdots + n^3 = n^3 \times n^3 = n^6$$

On a donc $f(n) = \mathrm{O}(n^6)$.
Considérons la deuxième moitié de la somme, les termes allant de $\left\lceil \frac{n^3+1}{2} \right\rceil$ à n^3. On a :

$$1 + 2 + 3 + \cdots + n^3 \geq \left\lceil \frac{n^3+1}{2} \right\rceil + \cdots + n^3 - 1 + n^3$$

Chacun des $\left\lceil \frac{n^3}{2} \right\rceil$ termes de droite est supérieur à $\left\lceil \frac{n^3+1}{2} \right\rceil$. On a donc :

$$f(n) \geq \left\lceil \frac{n^3}{2} \right\rceil \cdot \left\lceil \frac{n^3+1}{2} \right\rceil \geq \frac{n^3}{2} \cdot \frac{n^3}{2} = \frac{n^6}{4}$$

$f(n) = \Omega(n^6)$
$f(n) = \mathrm{O}(n^6)$ et $f(n) = \Omega(n^6)$, donc $f(n) = \Theta(n^6)$.

Index des commentaires

1.1 Bien vu!