

TD2 Récursivité et Chaines récursives

Arthur Garnier

February 4, 2015

1 Exercice 1

1.1

- $f(0)=0$
- $f(1)=1$
- $f(2)=1$
- $f(3)=2$
- $f(4)=3$
- $f(5)=5$

1.2

Suite de Fibonacci

1.3

$O(n) \rightarrow n$ appels récursifs

2 Exercice 2

2.1 Méthodologie

- Quel paramètre porte la récursion ?
- Quels sont les cas triviaux permettant une résolution directe ?
- Quel est le traitement général (comment faire pour n si on sait faire pour $n-1$?)?

2.2 Question 1

Longueur : Chaine $\rightarrow \mathbb{N}$, retourne le nombre de caractères dans la chaine

```
Si ch=chvide retourne 0
Sinon longueur(reste(ch))+1
```

Terminaison : Longueur de la chaine strictement décroissante lors des appels récursifs (reste(ch)) et arrêt à chvide

2.3 Question 2 : `est_membre(ch,c)`

```
si ch=chvide alors faux
sinon si premier(ch)=c alors vrai
sinon est_membre(reste(ch),c)
```

Terminaison : Même raisonnement que Q1

Complexité :

- Meilleur cas $O(1)$
- Cas moyen : $\frac{n}{2}$ si équiprobabilité des lettres et chaîne aléatoires.

2.4 Question 3 : `occurrence(chaine, c)`

```
si ch=chvide retourne 0
sinon si premier(ch)=c alors retourne 1+occurrence (reste(ch),c)
sinon retourne occurrence(reste(ch),c)
```

2.5 Question 4 : `tous_différents(chaine)`

```
si ch=chvide alors retourne vrai
sinon si est_membre(reste(ch), premier(ch)) alors retourne faux
sinon tous_différents(reste(ch))
```

Complexité : $O(n)*O(n) = O(n^2)$

2.6 Question 5 : `Supprime (ch,c)`

```
si ch=chvide alors retourne ch
sinon si premier(ch)=c
alors retourne reste(ch)
sinon adj(premier(ch), supprime(reste(ch),c)
```

2.7 Question 7 : `dernier(ch) → c`

précondition : $ch \neq chvide$

```
si reste(ch)=chvide alors retourne premier(ch)
sinon retourne dernier(reste(ch))
```

Complexité $O(n)$

2.8 Question 8 : `sauf_dernier(ch) → ch`

précondition : $ch \neq chvide$

```
si suite(ch) = chvide retourne chvide
sinon retourne adj(premier(ch), sauf_dernier(reste(ch))
```

2.9 Question 9 : nième(ch, n) → c

```
si n=0 retourne premier(ch)
sinon retourne meme(reste(ch), n-1)
```

2.10 Question 10 : npremiers(ch, n)

Précondition : $n \leq \text{longueur}(\text{ch})$

```
si n=1 ou n=0 retourne premier(ch)
sinon retourne adj(premier(ch), npremiers(reste(ch)), n-1))
```

Complexité : $O(n)$

2.11 Question 11 : nderniers(ch, n)

Précondition : $n \leq \text{longueur}(\text{ch})$

```
retourne(npremiers(retourne(ch, n))
```

ou

```
si longueur=n alors retourne ch
sinon retourne nderniers(reste(ch), n)
```

Complexité : $O(n) * O(n) = O(n^2)$

Meilleure complexité :

```
nderniers(ch, n) = ndernierbis(ch, longueur(ch) - n
```

```
nderniersbis(ch, k)
```

```
si k=0 alors retourne ch
sinon retourne nderniersbis(reste(ch), k-1)
```

2.12 Question 12 retourne(ch)

```
si ch=chvide alors return chvide
sinon return adj(dernier(ch), retourne(saufdernier(ch)))
```

```
retourne(ch) = retourne(ch, chvide)
```

```
retourne2(ch, ch_done)
si ch=chvide alors retourne ch_done
sinon return retourne retourne2(reste(ch), adj(premier(ch), ch_done)
```

- $O(n)$ en calcul
- $2n$ en mémoire

2.13 Question 13 : concat(ch1, ch2)

si ch1=chvide alors retourne ch2
sinon retourne adj(dernier(ch1), concat(saufdernier(ch1),ch2))

Complexité : $O(n^2)$