TP 1

Judicaël Courant

5 septembre 2016

1 Recherche dans un tableau quelconque

Écrire une fonction chercheindex(t, x). Appliquée à un tableau t et une valeur x, cette fonction devra retourner un index i tel que t[i] == x s'il en existe un, et la valeur -1 sinon.

2 Recherche dans un tableau trié

Écrire une fonction chercheindexord(t, x). Cette fonction devra comme précédemment retourner un index i tel que t[i] == x s'il en existe un, et la valeur -1 sinon. Cette fois-ci on suppose en outre que les valeurs du tableau t sont triées par ordre croissant.

Cette fonction devra travailler par dichotomie sur le tableau.

3 Recherche d'un zéro par dichotomie

Écrire une fonction cherchezero(f, a, b, eps) prenant en argument une fonction f, deux réels a et b vérifiant $a < b$ et un réel eps strictement positif.

On suppose de plus que f est continue sur l'intervalle $[a, b]$ et qu'on a

$$f(a) \leq 0 < f(b) \tag{1}$$

La fonction doit retourner une approximation à eps près d'un point où f s'annule.

Généraliser au cas où la propriété (1) est remplacée par la simple hypothèse que f(a) et f(b) sont de signes distincts.

4 Recherche d'un mot dans une chaîne de caractères

Écrire une fonction cherchemot(s, t) prenant en argument deux chaînes de caractères s et t et retournant un couple d'index (i, j), s'il en existe un, tel que t[i:j] == s. Quelles est la complexité de votre algorithme en fonction des longueurs n et p de t et s.

5 Exercice

Cet exercice est tiré du sujet 0 du concours CCP Math MP (publié en 2014). Des remarques et questions ont été rajoutées en italique.

Le code doit être écrit en langage Python (ou Scilab). En particulier, on prendra soin de bien respecter l'indentation.

1. Écrire une fonction factorielle qui prend en argument un entier naturel n et renvoie $n!$ (on n'acceptera pas bien sûr de réponse utilisant la propre fonction factorielle du module math de Python ou Scilab).
2. Écrire une fonction seuil qui prend en argument un entier M et renvoie le plus petit entier naturel n tel $n! > M$.
3. Écrire une fonction booléenne nommée est_divisible, qui prend en argument un entier naturel n et renvoie True si $n!$ est divisible par $n+1$ et False sinon.
4. On considère la fonction suivante nommée mystere :

```
def mystere(n) :
  s = 0
  for k in range(1,n+1) :
    s = s + factorielle(k)
  return s
```

   a) Quelle valeur renvoie mystere(4) ? *Essayez de le faire d'abord **sans** ordinateur. Puis avec ordinateur pour comprendre ce qu'il se passe.*

   b) Déterminer le nombre de multiplications qu'effectue mystere(n). *À faire sans ordinateur si possible. Mais pour vous aider, vous pouvez ajouter, dans la fonction factorielle, des instructions pour imprimer, à chaque appel, le nombre de multiplications faites.*

   c) Proposer une amélioration du script de la fonction mystere afin d'obtenir une complexité linéaire. *On écrira une fonction `mystere2` avec cette complexité et on vérifiera qu'on peut par exemple calculer `mystere(100000)`.*