TP 05

Algorithme dichotomique

Savoirs et compétences :

- ❐ *AA.C9 : Choisir un type de données en fonction d'un problème à résoudre*
- ❐ *AA.S11 : Manipulation de quelques structures de données.*
- ❐ *AA.S12 : Fichiers*

Consignes

Attention : suivez précisément ces instructions.

Votre fichier portera un nom du type

`tp05_durif_berne.py`,

où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.

- Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
- Commencez chaque question par son numéro écrit en commentaires.
- Les questions demandant une réponse écrite seront rédigées en commentaires.
- Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

Les figures seront sauvegardées sous le nom `tp05_Qnum_nom1_nom2.png`, où Qnum est le numéro de la question, et nom1, nom2 les noms des membres du binôme.

Première approche de la complexité

Étant donné deux suites (u_n) et (v_n) de réels strictement positifs, on dit que (u_n) est dominée par la suite (v_n) lorsque $\left(\dfrac{u_n}{v_n}\right)$ est une suite bornée. On note alors $u_n \underset{n\to+\infty}{=} O(v_n)$.

Par exemple :

- si (u_n) converge alors $u_n \underset{n\to+\infty}{=} O(1)$. Réciproque fausse;
- $3n \underset{n\to+\infty}{=} O(n^2)$, $5n^2 \underset{n\to+\infty}{=} O(n^2)$, $\ln(n) \underset{n\to+\infty}{=} O(n\ln(n)^2)$, $a n^2 + b n + c\ln(n) \underset{n\to+\infty}{=} O(n^2)$ ...
- pour tout polynôme de degré p, $P = a_p x^p + a_{p-1} x^{p-1} + ... + a_1 x + a_0$, on a $P(n) \underset{n\to+\infty}{=} O(n^p)$.

En programmation, on dira qu'un programme a :

- une complexité linéaire lorsque le nombre d'opérations effectuées est en $\underset{n\to+\infty}{O}(n)$;
- une complexité logarithmique lorsque le nombre d'opérations effectuées est en $\underset{n\to+\infty}{O}(\log(n))$;
- une complexité quadratique lorsque le nombre d'opérations effectuées est en $\underset{n\to+\infty}{O}(n^2)$.

Activité 1 – Recherche dichotomique dans un tableau trié

Recherche d'un élément dans une liste triée

On se donne une liste `L` de nombres de longueur `n`, triée dans l'ordre croissant, et un nombre `x0`.

Pour chercher `x0`, on va couper la liste en deux moitiés et chercher dans la moitié de tableau qui encadre `x0` et ainsi de suite...

On appelle `g` l'indice de l'élément du début de la sous-liste dans laquelle on travaille et `d` l'indice de l'élément de fin.

Au début, `g = 0` et `d = n - 1`.

On utilise la méthode suivante :

- on compare `x0` à « l'élément du milieu » `L[m]` avec `m = (g + d) // 2`;

- si `x0 = L[m]`, on a trouvé `x0`, on peut alors s'arrêter;
- si `x0` < `L[m]`, c'est qu'il faut chercher entre `L[g]` et `L[m-1]` ;
- si `x0` > `L[m]`, c'est qu'il faut chercher entre `L[m+1]` et `L[d]`.

On poursuit jusqu'à ce qu'on a trouvé `x0` ou lorsque l'on a épuisé la liste `L`.

Question 1 *Illustrer la méthode avec les deux exemples suivants :*

- `x0 = 5` et `L` =

−3	5	7	10	11	14	17	21	30

- `x0 = 11` et `L` =

−2	1	2	7	8	10	13	16	17

.

Question 2 *Si* `x0` *n'est pas dans la liste* `L`*, donner un test d'arrêt du processus de dichotomie portant sur g et d.*

Question 3 *Écrire une fonction* `dichotomie(x0,L)` *qui renvoie* `True` *ou* `False` *selon que* `x0` *figure ou non dans* `L` *par cette méthode. On utilisera une boucle* `while` *que l'on interrompra soit lorsque l'on a trouvé* `x0`*, soit lorsque l'on a fini de parcourir la liste.*

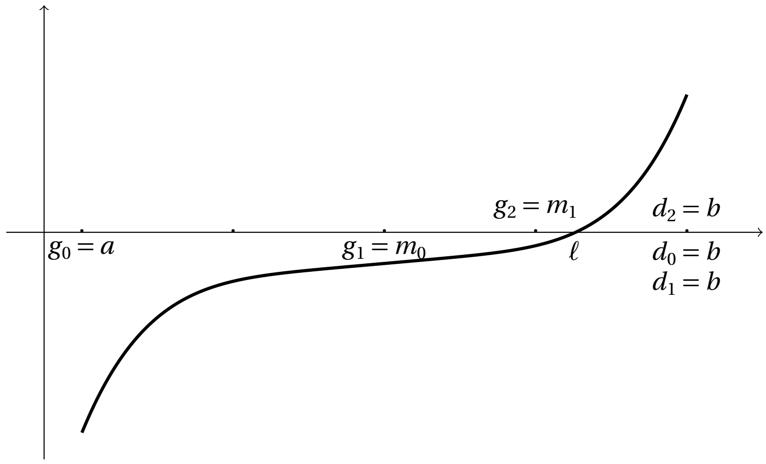
Question 4 *Combien vaut $g-d$ au i^e tour de boucle ? Si* `x0` *ne figure pas dans* `L`*, montrer que le nombre de tours de boucles nécessaires pour sortir de la fonction est de l'ordre de* $\ln n$ *où n=len(L) (cela rend la fonction beaucoup plus efficace qu'une simple recherche séquentielle pour laquelle le nombre de comparaisons pour sortir de la boucle serait de l'ordre de n).*

Activité 2 – Recheche d'un zéro d'une fonction

Recherche d'un zéro d'une fonction

Soit une fonction $f:[a,b]\rightarrow\mathbb{R}$ $(a<b)$ vérifiant : f continue sur $[a,b]$ et $f(a)\cdot f(b)\leqslant 0$ ie $f(a)$ et $f(b)$ de signes opposés.

Le théorème des valeurs intermédiaires s'applique et assure que f possède au moins un zéro ℓ entre a et b.



L'idée consiste à créer une suite d'intervalles $[g_n,d_n]$ tels que pour tout entier naturel n,

$$g_n\leqslant\ell\leqslant \mathrm{d}_n \text{ et } 0\leqslant g_n-d_n=\frac{g_{n-1}-d_{n-1}}{2}.$$

On considère $m_0=\dfrac{g_0+d_0}{2}$ et on évalue $f(m_0)$:

- Si $f(m_0)\times f(d_0)\geqslant 0$, on va poursuivre la recherche d'un zéro dans l'intervalle $[g_1,d_1]=[g_0,m_0]$.
- Sinon, on poursuit la recherche dans l'intervalle $[g_1,d_1]=[m_0,d_0]$.
- On recommence alors en considérant $m_1=\dfrac{g_1+d_1}{2}$ ...

Question 5 *Si l'on souhaite que g_n et d_n soient des solutions approchées de ℓ à une précision ε, quelle est la condition d'arrêt de l'algorithme ? Préciser alors la valeur approchée de ℓ qui sera renvoyée par la fonction.*

Question 6 *Écrire une fonction* `recherche_zero(f,a,b,epsilon)` *qui renvoie une valeur approchée du zéro de* `f` *sur* `[a,b]` *a epsilon près.*

Question 7 *Tester la fonction avec* $f : x \mapsto x^2 - 2$ *sur* $[0,2]$ *et* $\varepsilon = 0,001$.

Question 8 *Avec une erreur de* $\varepsilon = \dfrac{1}{2^p}$, *combien y a-t-il de comparaisons au final en fonction de p ?*