

Exercices 1

Philipp Ahrendt

November 28, 2025

Contents

1 Exercice 1: Années bissextiles	1
2 Exercice 2: Algorithme d'Euclide	1
2.1 Question a:	2
2.2 Question b:	2
3 Exercice 3: Recherche de fonctions inverses	2
3.1 Question a: Inversion sur les entiers	2
3.2 Question b: Recherche dichotomique	3
3.3 Question c: Recherche dichotomique sur les nombres réels . .	4

1 Exercice 1: Années bissextiles

Les années bissextiles (les années avec 29 jours en février au lieu de 28) sont les années qui respectent l'une des conditions suivantes:

1. L'année est divisible par 4 sans être divisible par 100.
2. L'année est divisible par 400.

Écrire une fonction `bissextile` qui prend en entrée un entier (représentant l'année) et retourne `True` si l'année est bissextile, et `False` sinon.

2 Exercice 2: Algorithme d'Euclide

Le plus grand diviseur commun de deux entiers non nuls n, m , noté $\text{pgcd}(n, m)$, est le plus grand entier k tel que n/k et m/k soient tous les deux entiers. Les deux propriétés suivantes sont vérifiées par le `pgcd`:

1. Si m divise n , alors $\text{pgcd}(n,m)=m$.
2. Sinon, $\text{pgcd}(n,m)=\text{pgcd}(m,n \% m)$, où $n \% m$ est le reste de la division euclidienne.

2.1 Question a:

Utiliser les deux propriétés ci-dessus pour écrire une fonction `pgcd`, qui prend en entrée deux entiers, et donne en sortie leur `pgcd`.

2.2 Question b:

Pouvez-vous expliquer pourquoi cette fonction finit toujours par rendre un résultat (i.e pourquoi elle ne peut jamais tourner à l'infini)? Pouvez-vous expliquer pourquoi elle calcule bien le `pgcd`?

3 Exercice 3: Recherche de fonctions inverses

Si $f(x)$ est une fonction strictement croissante en x , elle admet une fonction inverse $g(y)$ telle que $g(f(x))=x$ et $f(g(y))=y$. Dans cet exercice, le but est de trouver algorithmiquement des approximations de la fonction inverse d'une fonction f croissante, par différentes méthodes.

3.1 Question a: Inversion sur les entiers

Une première approche consiste à chercher les valeurs de la fonction inverse sur les entiers uniquement (positifs et négatifs). Pour une fonction f strictement croissante, on peut définir son inverse g sur les entiers par:

- $g(n)=k$, où k est le plus grand entier tel que $f(k) \leq n$

Écrire une fonction `inv_int` qui prend en entrée une fonction f (que l'on suppose strictement croissante), et un entier n , et donne en résultat la valeur $g(n)$.

Pour tester cette fonction, vous pouvez utiliser par exemple $f(x) = x^3$, et tester contre la racine cubique.

```
import math

def f(n):
    return n**3
```

```

def g_int(n):
    # La fonction que doit trouver l'algorithme
    return math.floor(n**(1/3))

# Tester sur cette fonction aussi. L'algorithme marche-t-il encore?
def f_bis(n):
    return (n+5)**3

def g_bis_int(n):
    return math.floor(n**(1/3)-5)

```

3.2 Question b: Recherche dichotomique

Une manière de calculer efficacement l'inversion entière d'une fonction est par **recherche dichotomique**. Dans un premier temps, on va supposer que la fonction f vérifie $n \leq f(n)$. Le but est toujours de trouver $g(n)$:

- $g(n)=k$, où k est le plus grand entier tel que $f(k) \leq n$.

L'algorithme de recherche dichotomique consiste à diminuer progressivement l'intervalle dans lequel on cherche $g(n)$. A chaque étape, on a un intervalle $[k_{\min}, k_{\max}]$, et on prend k dans cet intervalle. Si k est plus grand que $g(n)$, on remplace k_{\max} par k , et on relance la recherche dans $[k_{\min}, k]$, et si k est plus petit, on remplace k_{\min} et relance. On répète cela jusqu'à ce qu'on trouve $g(n)$. Voici une manière d'implémenter cela:

1. Initialiser:

- $k_{\min} \leftarrow 0$
- $k_{\max} \leftarrow n$

2. Tant que $g(n)$ n'est pas trouvée:

- $k \leftarrow$ partie entière de $(k_{\min} + k_{\max})/2$
- Si $f(k) \leq n$:
 - Si $f(k+1) > n$, **fin**: on sort $g(n)=k$
 - Sinon, $k_{\min} \leftarrow k$
- Si $f(k) > n$, $k_{\max} \leftarrow k$

Écrire une fonction `inv_int_dicho` qui prend une fonction `f` et un entier `n`, et trouve l'inverse `g(n)` par recherche dichotomique.

Comment peut-on adapter cette fonction pour qu'elle marche sur n'importe quelle fonction croissante?

3.3 Question c: Recherche dichotomique sur les nombres réels

On peut utiliser le même principe pour approcher l'inverse sur les nombres réels. Une fonction inverse approchée d'une fonction $f(x)$ est une fonction $g(y)$ telle que $|g(f(x)) - x| \leq \epsilon$, pour une tolérance d'erreur $\epsilon > 0$ choisie au préalable.

Adapter l'algorithme précédent pour écrire une fonction `inv_dicho`, qui prend en entrée une fonction `f`, un nombre `x`, et un nombre positif ϵ , et trouve $g(x)$ avec une erreur maximale de ϵ .