Exercices

Exercice ITC2.1 : Tri à bulles [*]

On dispose d'une liste L de nombres ; on veut la trier dans l'ordre croissant. On propose l'algorithme suivant :

```
1  n ⟵ longueur de L
2  pour i allant de 1 à n-1 faire
3  |  j ⟵ i
4  |  x ⟵ L[i]
5  |  tant que j>0 et L[j-1]>x faire
6  |  |  L[j] ⟵ L[j-1]
7  |  |  j ⟵ j-1
8  |  fin tq
9  |  L[j] ⟵ x
10 fin pour
```

1. Prouvez que la boucle Tant que se termine.
2. On considère une liste $L = [4, 7, 2, 3, 1]$. Exécutez à la main cet algorithme sur la liste L en donnant ce que vaut L à la fin de chaque itération de la boucle principale.
3. Proposez alors (sans démonstration) un invariant de cette boucle permettant de prouver que la liste est bien triée à la fin.

Exercice ITC2.2 : Somme des termes d'une liste [**]

L'algorithme suivant calcule la somme des termes d'une liste L, c'est-à-dire $\sum_{k=0}^{len(L)-1} L[k]$:

```
1  somme ⟵ 0
2  i ⟵ 0
3  n ⟵ longueur de L
4  tant que i<n faire
5  |  i ⟵ i+1
6  |  somme ⟵ somme+L[i]
7  fin tq
8  retourner somme
```

Dans le suite on notera n la longueur de la liste L.

1. Montrez que l'algorithme se termine à l'aide de la fonction de terminaison $T = n - i$.
2. Montrez que
$$I : somme = \sum_{k=0}^{i-1} L[k] \text{ et } i \leq n$$
n'est pas un invariant de cette boucle ; on pourra se contenter de regarder ce qui se passe à la fin du premier passage dans la boucle.
3. Corrigez l'algorithme, et montrez que maintenant I est un invariant de la boucle.
4. Déduisez-en que l'algorithme renvoie bien la valeur souhaitée.

Exercice ITC2.3 : Produit de deux entiers positifs [**]

On considère la fonction suivante en Python, qui prend deux entiers a et b positifs :

```
1  def produit(a,b):
2      p=0
3      q=b
```

```
    while q>0:
        p=p+a
        q=q-1
    return p
```

1. Montrez que la boucle se termine à l'aide de la fonction $T = q$.
2. Montrez que la proposition
$$I : p = a \times (b - q) \text{ et } q \geq 0$$
est un invariant de la boucle.
3. Déduisez-en que la fonction renvoie bien le produit de a par b.

On propose un autre algorithme plus compliqué :

```
def produit2(a,b):
    p=0
    puissance=a
    q=b
    while q>0:
        if q%2==1:
            p=p+puissance
        puissance=puissance*2
        q=q//2
    return p
```

4. Démontrez que cet algorithme se termine.
5. Montrez que $J : \{p + puissance * q = a * b \text{ et } q \geq 0\}$ est un invariant de cette boucle.
6. Déduisez-en la correction de cet algorithme.

Exercice ITC2.4 : PGCD de deux entiers strictement positifs [***]

Soient a et b deux entiers strictement positifs. Le PGCD de a et b est le plus grand entier qui divise à la fois a et b. Par exemple, $PGCD(6, 8) = 2$.
On propose l'algorithme suivant pour calculer le PGCD de a et b :

```
u ⟵ a
v ⟵ b
tant que u est différent de v faire
    si u>v alors
        u ⟵ u-v
    sinon
        v ⟵ v-u
    fin si
fin tq
retourner u
```

1. Justifiez que u et v ne peuvent pas être nuls à la fin de la boucle. Déduisez-en qu'ils ne sont jamais nuls.
2. Montrez que la boucle se termine à l'aide de la fonction $T = max(u, v) - 1$
3. Montrez que
$$I : PGCD(a, b) = PGCD(u, v)$$
est un invariant de la boucle.
4. EN déduire la correction de cet algorithme.

Exercice ITC2.5 : Recherche par dichotomie dans un tableau trié [***]

On dispose d'une liste L triée par ordre croissant, et d'un nombre a présent dans la liste. On cherche à connaître l'indice d tel que $L[d] = a$. On propose la fonction Python suivante :

```python
def index_element(L,a):
    g=0
    d=len(L)-1
    while d>g:
        m=(d+g)//2
        if L[m]<a:
            g=m+1
        else:
            d=m
    return d
```

Si la liste ne contient que l'élément a, il est évident que l'algorithme fonctionne sans passer par la boucle, car initialement $g = d = 0$. On se place donc dans le cas où $len(L) \geq 2$.

1. Montrez que la boucle se termine à l'aide de la fonction $T = d - g$.
2. Montrez que
$$I : L[g] \leq a \leq L[d] \text{ et } g < d$$
est un invariant de la boucle.
3. Déduisez-en que l'algorithme est correct.

Corrections des exercices

Corrigé de l'exercice ITC2.1 : Tri à bulles [*]

1. Il est évident que $T = j$ est une fonction de terminaison de cette boucle.
2. À chaque tour de boucle, l'algorithme considère un élémente de plus (en commençant par le second) et le fait "remonter" dans la liste tant qu'il est plus petit que son terme précédent :

itération	L
1	[4,7,2,3,1]
2	[2,4,7,3,1]
3	[2,3,4,7,1]
4	[1,2,3,4,7]

3. Il semble bien que la propriété "La sous-liste comprenant les termes de la liste entre 0 et i est triée" soit un invariant de cette boucle. À la fin, $i = n - 1$ donc toute la boucle est triée.

Corrigé de l'exercice ITC2.2 : Somme des termes d'une liste [**]

1. T est à valeurs entières, décroît de 1 à chaque passage dans la boucle, et la boucle s'arrête lorsque $T \leq 0$ donc c'est une bonne fonction de terminaison.
2. Après un passage dans la boucle, on a $i = 1$ et $somme = L[1]$ au lieu d'avoir $somme = L[0]$ car on a incrémenté i avant d'ajouter $L[i]$ à $somme$.
3. On passe la ligne `i←i+1` après la ligne `somme←somme+L[i]`. Montrons alors que I est un invariant de la boucle. Tout d'abord, au début $i = 0 \leq n$ et $somme = 0 = \sum_{k=0}^{-1} L[k]$: c'est bon.
   Si on est dans la boucle, on sait déjà que $i < n$ au début donc $i \leq n$ à la fin car on a ajouté 1 à i. De plus, au début, $somme = \sum_{k=0}^{i_i-1} L[k]$ en notant i_i la valeur de i au début de la boucle et $i_f = i_i + 1$ à la fin. Au cours de la boucle on effectué $somme \to somme + L[i_i]$ donc à la fin $somme = \sum_{k=0}^{i_i} L[k] = \sum_{k=0}^{i_f-1} L[k]$ donc la proporiété reste vraie.
4. À la fin : $somme = \sum_{k=0}^{i-1} L[k]$ et $i \leq n$ et $i \geq n$ (terminaison de la boucle) donc $i = n$ et $somme = \sum_{k=0}^{n-1} L[k]$.

Corrigé de l'exercice ITC2.3 : Produit de deux entiers positifs [**]

1. T est une fonction à valeurs entières.
   À chaque passage dans la boucle, q diminue de 1 donc T est strictement décroissante.
   La boucle se termine lorsque $q \leq 0$ soit $T \leq 0$
   Donc T est une fonction de terminaison de la boucle, donc celle-ci se termine.
2. Initialisation : au départ, $q = b$ donc $a \times (b - q) = 0 = p$ et de plus $b \geq 0$ par hypothèse donc $q \geq 0$.
   Supposons que I soit vraie au début d'une itération. Notons p_i, q_i les valeurs au début de la boucle et p_f, q_f à la fin. Alors par hypothèse $p_i = a.(b - q_i)$ et de plus $q_i > 0$ puisque la condition de départ a été vérifiée, donc $q_i \geq 1$. À la fin, on a $q_f = q_i - 1$ d'où on tire $q_f \geq 0$. De plus, $p_f = p_i + a = a.(b - q_i) + a = a.(b - q_i + 1) = a.(b - q_f)$ puisque $q_f = q_i - 1$. On en déduit que la condition I reste vraie à la fin de la boucle, c'est donc un invariant de la boucle.
3. Une fois la boucle terminée, on a donc : $p = a.(b - q)$ et $q \geq 0$ et $q \leq 0$ (condition de sortie) donc $q = 0$ et $p = a.b$.
4. $T = q$ est une bonne fonction de terminaison (à valeurs entières car q subit une division entière, strictement décroissante car pour tout entier $q \geq 1$ on a $q//2 < q$, et $T \leq 0 \Rightarrow$ la boucle se termine).
5. Initialisation évidente avec les valeurs avant la boucle.
   Supposons que J soit vraie au début d'une itération. Notons $p_i, q_i, puissance_i$ les valeurs au début de la boucle et $p_f, q_f, puissance_f$ à la fin. Alors par hypothèse $p_i + puissance_i * q_i = a * b$ et $q_i > 0$ par la condition de boucle. On va séparer deux cas :
   — si q_i est pair, alors $\begin{cases} p_f & = p_i \\ q_f & = \frac{q_i}{2} \\ puissance_f & = puissance_i^2 \end{cases}$ donc $p_f + puissance_f * q_f = p_i + \left(puissance_i^2\right) * q_i/2 = p_i + puissance_i * q_i = a * b$ par hypothèse

— si q_i est impair, alors $\begin{cases} p_f & = p_i + puissance_i \\ q_f & = \frac{q_i - 1}{2} \\ puissance_f & = puissance_i^2 \end{cases}$ donc $p_f + puissance_f * q_f = p_i + puissance +$
$\left(puissance_i^2\right) * (q_i - 1)/2 = p_i + puissance_i + puissance_i * (q_i - 1) = a * b$ par hypothèse

De plus dans les deux cas $q_i > 0 \Rightarrow q_f \geq 0$. Donc J est vraie à la fin de l'itération.

Conclusion : J est un invariant de la boucle.

6. Après la boucle, on a donc $p + puissance * q = a * b$ et $q \leq 0$ et $q \geq 0$ donc $q = 0$ et $p = a * b$ ce qui est le résultat recherché.

Corrigé de l'exercice ITC2.4 : PGCD de deux entiers strictement positifs [***]

1. Si u est modifié dans la boucle, c'est pour prendre la valeur $u - v$, or si on est dans la boucle c'est que $u \neq v$ donc u ne peut pas prendre la valeur 0. Il en est de même pour v.

   Donc tout au long de l'algorithme, u et v seront des entiers strictement positifs.

2. T est à valeurs entières car u et v sont entiers.

   Pour la décroissance, soient u_i, v_i les valeurs initiales et u_f, v_f les valeurs finales. Supposons que $u_i > v_i$ alors $T_i = u_i - 1$ et à la fin on a $u_f = u_i - v_i < u_i$ (car $v_i > 0$) et $v_f = v_i < u_i$ donc forcément $max(u_f, v_f) - 1 < T_i$: la fonction T a donc diminué strictement. C'est pareil dans le cas $v_i > u_i$.

   Enfin, si $T \leq 0$ alors c'est que $max(u, v) = 1$ et comme u et v sont des entiers strictement positifs, c'est que $u = v = 1$ donc $|u - v| = 0$ donc la boucle se termine.

   Remarque : ici elle peut même se terminer avant que T atteigne la valeur 1.

3. Initialisation : au début $u = a$ et $v = b$ donc la propriété est évidente.

   Soient u_i, v_i les valeurs au début de la boucle et u_f, v_f à la fin. On suppose que $PGCD(a, b) = PGCD(u_i, v_i)$ qu'on va noter p. Cela signifie que p est le plus grand entier divisant a et b, et aussi u_i et v_i.

   Supposons que $u_i > v_i$, alors $v_f = v_i$ et $u_f = u_i - v_i$. Il est alors évident que tout nombre qui divise u_i et v_i divise aussi u_f et v_f, et vice-versa. Donc $PGCD(u_f, v_f) = PGCD(u_i, v_i)$ donc $PGCD(u_f, v_f) = PGCD(a, b)$. Il en va de même dans l'autre cas.

   Donc I est un invariant de la boucle.

4. À la fin, $u = v$ (sortie de la boucle) donc $PGCD(u, v) = u = v$ (évident) donc $PGCD(a, b) = u = v$.

Corrigé de l'exercice ITC2.5 : Recherche par dichotomie dans un tableau trié [***]

1. T est à valeurs entières. Il est évident que la boucle se termine lorsque $T \leq 0$. Reste à prouver qu'elle est strictement décroissante.

   Pour cela, on teste les deux cas. Soient d_i, g_i les valeurs au début de la boucle et d_f, g_f à la fin. Comme $g > d$ dans la boucle, il est évident que $g < \frac{d+g}{2} < d$ donc avec la partie entière : $g \leq m < d$. Si $L[m] < a$ alors à la fin on a $g_f = m + 1 \geq g_i + 1$ donc $g_f > g_i$ et $d_f = d_i$ donc $T_f < T_i$; dans l'autre cas on a $g_f = g_i$ et $d_f = m < d_i$ donc encore $T_f < T_i$.

   On en déduit que T est une fonction de terminaison, donc la boucle se termine.

2. Initialement, on sait par hypothèse que a se trouve dans la liste qui est triée par ordre croissant, donc $L[O] \leq a \leq L[len(L) - 1]$ donc la propriété est vraie.

   Supposons que la propriété soit vraie au début de la boucle : $L[g_i] \leq a \leq L[d_i]$ et $g_i < d_i$. Si $L[m] < a$ alors c'est que a se trouve à un indice strictement supérieur à m, donc $L[m + 1] \leq a$ soit $L[g_f] \leq a \leq L[d_f]$ donc la propriété est vraie à la fin. Si au contraire $L[m] \geq a$ alors a se trouve à un indice $\leq m$ donc $L[g_f] \leq a \leq L[d_f]$. Dans tous les cas la propriété est vraie à la fin.

   I est donc un invariant de la boucle.

3. À la fin, $g \geq d$ et $L[g] \leq a \leq L[d]$. Comme le tableau est trié par ordre croissant, $L[g] \leq L[d] \Rightarrow g \leq d$. On en déduit que $g = d$ et donc $L[g] = a = L[d]$.