|  |
| --- |
| **Correction activité : Recherche d’une occurrence** |

Une fois les algorithmes faits, on peut compter le nombre d’opérations (affectations, opérations arithmétiques, testes, itérations d’une boucle, affichage, renvois d’une fonction, etc) pour montrer que la complexité asymptotique des algorithmes utilisés est linéaire : *O(n).*

**Situation 1**

def occurence\_1(L):

n=len(L) *# (1 affectation)*

i=0 #on initialise i à zéro *(1 affectation)*

while ((i<n) and L[i]!=40): *# (n itérations + 2 testes / itération au pire des cas)*

i=i+1 #on incrémente i tant qu’on ne trouve pas le joueur gagnant *(1 affectation + 1 addition)*

if (i<n): #dans ce cas on trouve l’indice du joueur gagnant *(1 teste)*

print(i+1, "est le numéro du joueur gagnant") *# (1 affichage + 1 addition)*

else: #dans ce cas on a parcouru toute la liste L *(1 teste)*

print ("Aucun joueur a fait un score de 40") *# (1 affichage)*

Total :

**Situation 2**

def occurence\_2(L,x):

n=len(L)

i=0

while ((i<n) and L[i]!=x):

i=i+1

if (i<n):

print(i+1,"est le numéro du joueur gagnant")

else:

print ("Aucun joueur a fait un score de",x)

L’algorithme étant pratiquement le même que précédemment, puisqu’on a remplacé 40 par x, alors la complexité asymptotique est la même : *O(n)*.

**Preuve de terminaison**

On peut se contenter de prouver la terminaison et la correction partielle de la fonction *occurrence\_1()* puisque la fonction *occurrence\_2()* est basée sur le même algorithme.

Pour prouver la terminaison de la boucle « while », il suffit que les conditions de la boucle ne soient plus vérifiées. Ici, on arrête la boucle s’il existe un ***i*** compris entre ***0*** et ***n-1*** tel que ***L[i]=40*** (dans le meilleur des cas) ou bien si ***i*** est incrémenté jusqu’à atteindre la valeur ***n*** (dans le pire des cas). Le variant est donc ***n-i*** qui varie de ***n*** à ***0*** et garantie que la boucle se termine.

**Correction partielle**

**Précondition**: et pour tout

**Postcondition**: Trouver ***i*** tel ***L[i]=40*** ou affiche un message pour dire que ***40*** n’est pas dans la liste ***L***.

**Invariant**:

Preuve de la terminaison et correction partielle montre ainsi la correction total de l’algorithme.