
TP 5 : Tris de listes

2 BCPST 2 - TP Informatique

```
# Q1
import time

def insert_element(L, a):
    if L == [] or L[0] >= a:
        return [a] + L

    return [L[0]] + insert_element(L[1:], a)

# Q2

def tri_insertion(L):
    if L == []:
        return L

    return insert_element(tri_insertion(L[1:]), L[0])

L = [9, 8, 7, 6, 5, 4, 3, 2, 1]

print(tri_insertion(L))

# %% Test vitesse

N = 200
L = [N-i for i in range(N)]
t1 = time.time()
for i in range(100):
    tri_insertion(L)
t2 = time.time()
print("temps de tri par insertion sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
L = [i for i in range(N)]
for i in range(100):
    tri_insertion(L)
t2 = time.time()

print("temps de tri par insertion sur liste triée :")
print(t2-t1, "secondes")
```

```

# %%
# Q3

def tri_bulle(L):
    n = len(L)
    again = True
    while again:
        again = False
        for i in range(n-1):
            if L[i] > L[i+1]:
                L[i], L[i+1] = L[i+1], L[i]
                again = True
    return L

L = [9, 8, 7, 6, 5, 4, 3, 2, 1]
print(tri_bulle(L))

# %% Test vitesse
N = 100
t1 = time.time()
for i in range(N):
    L = [N-i for i in range(N)]
    tri_bulle(L)
t2 = time.time()

print(t2-t1)

t1 = time.time()
for i in range(N):
    L = [i for i in range(N)]
    tri_bulle(L)
t2 = time.time()

print(t2-t1)
# %%

# Q4-Q5

def tri_selection(L):
    n = len(L)
    Lt = []
    for i in range(n):
        m = min(L)
        L.remove(m)
        Lt.append(m)
    return Lt

L = [9, 8, 7, 6, 5, 4, 3, 2, 1]
print(tri_selection(L))
print(L)

```

```

# %% Test vitesse
N = 200
t1 = time.time()
for i in range(N):
    L = [N-i for i in range(N)]
    tri_selection(L)
t2 = time.time()

print("temps de tri par selection sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
for i in range(N):
    L = [i for i in range(N)]
    tri_selection(L)
t2 = time.time()

print("temps de tri par selection sur liste triée :")
print(t2-t1, "secondes")

# %%
# Q5

def liste_vers_C(L):
    a = min(L)
    b = max(L)
    C = []
    for i in range(b-a+1):
        C.append(0)

    for i in L:
        C[i-a] += 1
    return C

print(liste_vers_C([1, 1, 2, 2, 5, 3, 6]))

# Q6

def tri_comptage(L):
    C = liste_vers_C(L)
    a = min(L)
    il = 0
    for i in range(len(C)):
        while C[i] > 0:
            C[i] -= 1
            L[il] = i+a
            il += 1
    return L

L = [9, 8, 7, 6, 5, 4, 3, 2, 1]
print(tri_comptage(L))
print(L)

```

```

# %% Test vitesse
N = 200
t1 = time.time()
for i in range(N):
    L = [N-i for i in range(N)]
    tri_comptage(L)
t2 = time.time()

print(t2-t1)

t1 = time.time()
for i in range(N):
    L = [i for i in range(N)]
    tri_comptage(L)
t2 = time.time()

print(t2-t1)

# %%
# Tri Fusion

# %%
# Q6

def separe(L):
    n = len(L)
    p = (n+1)//2
    return L[:p], L[p:]

L = [0, 1, 2, 3, 4, 5, 6, 7, 8]

# Q7

def fusion(L1, L2):
    if len(L1) == 0 or len(L2) == 0:
        return L1 + L2
    if L1[0] > L2[0]:
        return [L2[0]] + fusion(L1, L2[1:])
    else:
        return [L1[0]] + fusion(L1[1:], L2)

# Q8

def tri(L):
    if len(L) == 1:
        return L
    L1, L2 = separe(L)
    return fusion(tri(L1), tri(L2))

```

```

# %%

N = 100
t1 = time.time()
for i in range(N):
    L = 3*[N-i for i in range(N)]
    tri_insertion(L)
t2 = time.time()

print("temps de tri par insertion sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
for i in range(N):
    L = 3*[N-i for i in range(N)]
    tri_selection(L)
t2 = time.time()

print("temps de tri par selection sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
for i in range(N):
    L = 3*[N-i for i in range(N)]
    tri_bulle(L)
t2 = time.time()

print("temps de tri à bulle sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
for i in range(N):
    L = 3*[N-i for i in range(N)]
    tri_comptage(L)
t2 = time.time()

print("temps de tri par comptage sur liste non triée :")
print(t2-t1, "secondes")

t1 = time.time()
for i in range(N):
    L = 3*[N-i for i in range(N)]
    tri(L)
t2 = time.time()

print("temps de tri par fusion sur liste non triée :")
print(t2-t1, "secondes")

```

```

# %%
#Q12
def plus_grand_mot(s1, s2):

    n = min(len(s1), len(s2))
    for i in range(n):

        if ord(s1[i]) > ord(s2[i]):
            return s1
        elif ord(s1[i]) < ord(s2[i]):
            return s2

    if (len(s1) > len(s2)):
        return s1
    else:
        return s2

print(plus_grand_mot("lac", "linge"))

#Q13
# Avec le tri à bulles :

def tri_alphabétique(L):

    triee = False

    while(not(triee)):
        triee = True
        for i in range(len(L)-1):

            if(L[i] != L[i+1] and plus_grand_mot(L[i], L[i+1]) == L[i]):
                L[i], L[i+1] = L[i+1],L[i]
                triee = False

    return L

L = ["do","re","mi","fa","sol","la","si","do"]

print(tri_alphabétique(L))

```