Treball de Recerca: Algoritmes d'ordenació

Joan Coma Bages

Curs 2020/2021

Índex

1. Introducció

2. Què és ordenar un vector?

3. Selection sort

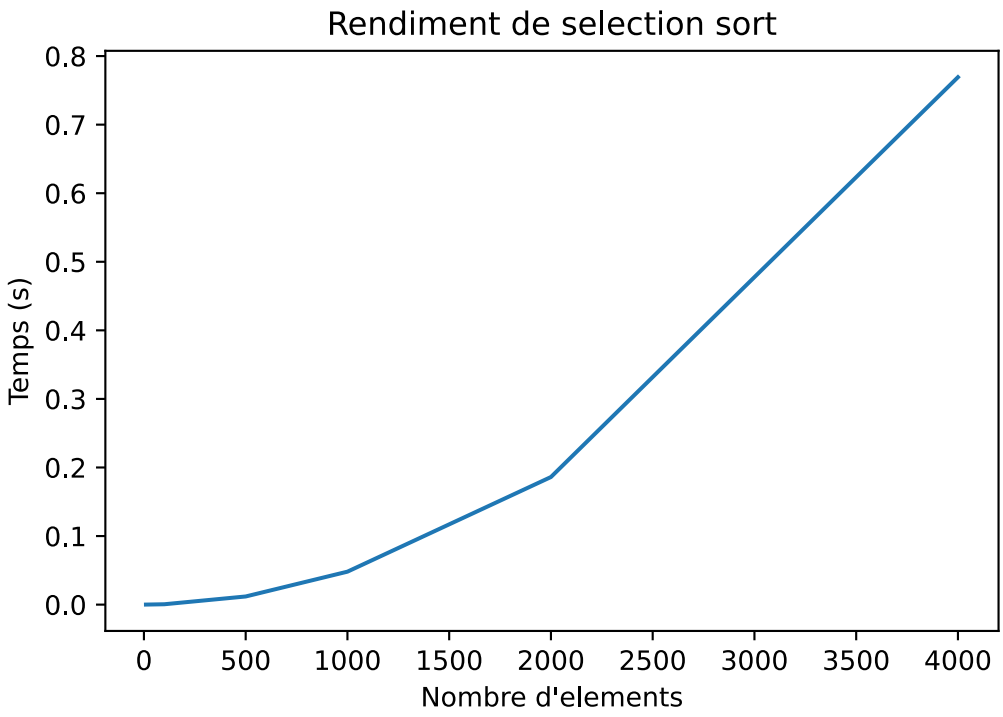
3.1. Com funciona?

3.2. Pseudocodi

3.3. Implementació

```
#!/usr/bin/env python3
import utils

def sort(array):
    for i in range(len(array[:-1])):
        low = i
        for j in range(i, len(array)):
            if array[low] > array[j]:
                low = j

        array[i], array[low] = array[low], array[i]
    return array

if __name__ == "__main__":
    array = utils.numbers()
    print(sort(array))
```

3.4. Rendiment



4. Insertion sort

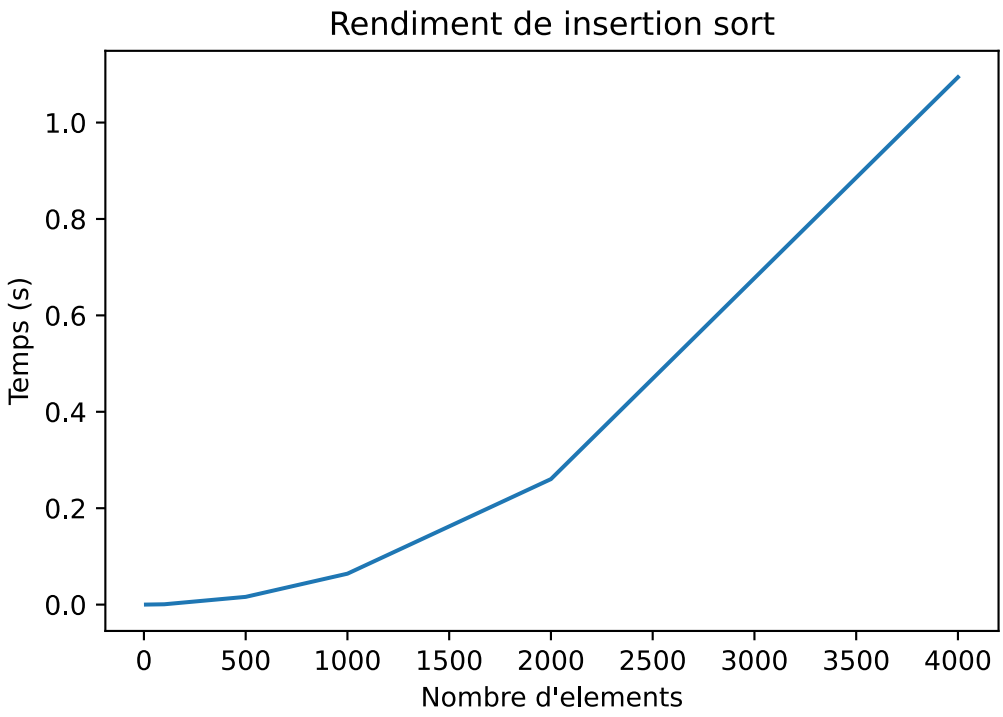
4.1. Com funciona?

4.2. Pseudocodi

4.3. Implementació

```python
#!/usr/bin/env python3
import utils

def sort(array):
    for i in range(len(array)):
        for j in range(len(array[:i])):
            if array[i] < array[j]:
                array[i], array[j] = array[j], array[i]
    return array

if __name__ == "__main__":
    array = utils.numbers()
    print(sort(array))
```

4.4. Rendiment



5. Bubble sort

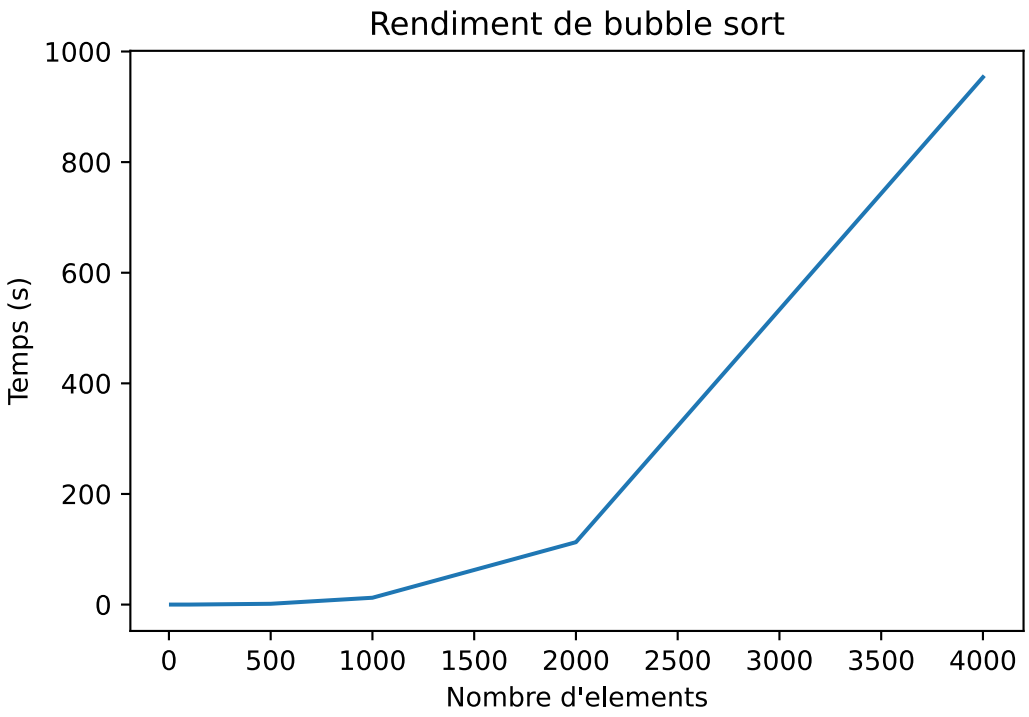
5.1. Com funciona?

5.2. Pseudocodi

5.3. Implementació

```
#!/usr/bin/env python3
import utils

def sort(array):
    while True:
        for i in range(len(array[:-1])):
            if array[i] > array[i+1]:
                array[i], array[i+1] = array[i+1], array[i]
                break

        # aquest else s'executa si el bloc anterior s'ha executat sense cap break, és a dir, si mai es compleix la condició array[i] > array[i+1]
        else:
            break
    return array

if __name__ == "__main__":
    array = utils.numbers()
    print(sort(array))
```

5.4. Rendiment



6. Shell sort

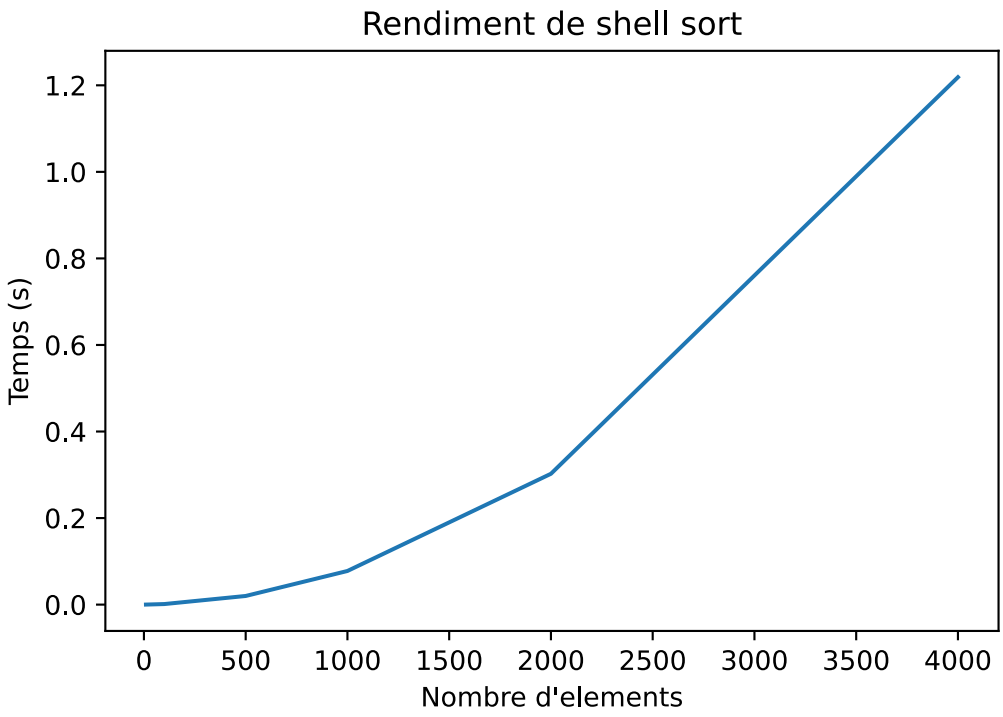
6.1. Com funciona?

6.2. Pseudocodi

6.3. Implementació

```python
#!/usr/bin/env python3
import utils
import insertion

def sort(array):
    step = len(array) // 2
    while step != 0:
        for offset in range(step):
            sorted = insertion.sort(array[offset::step])
            for i in range(len(sorted)):
                array[step*i + offset] = sorted[i]
        step //= 2
    return sorted

if __name__ == "__main__":
    array = utils.numbers()
    print(sort(array))
```

6.4. Rendiment



7. Quick sort

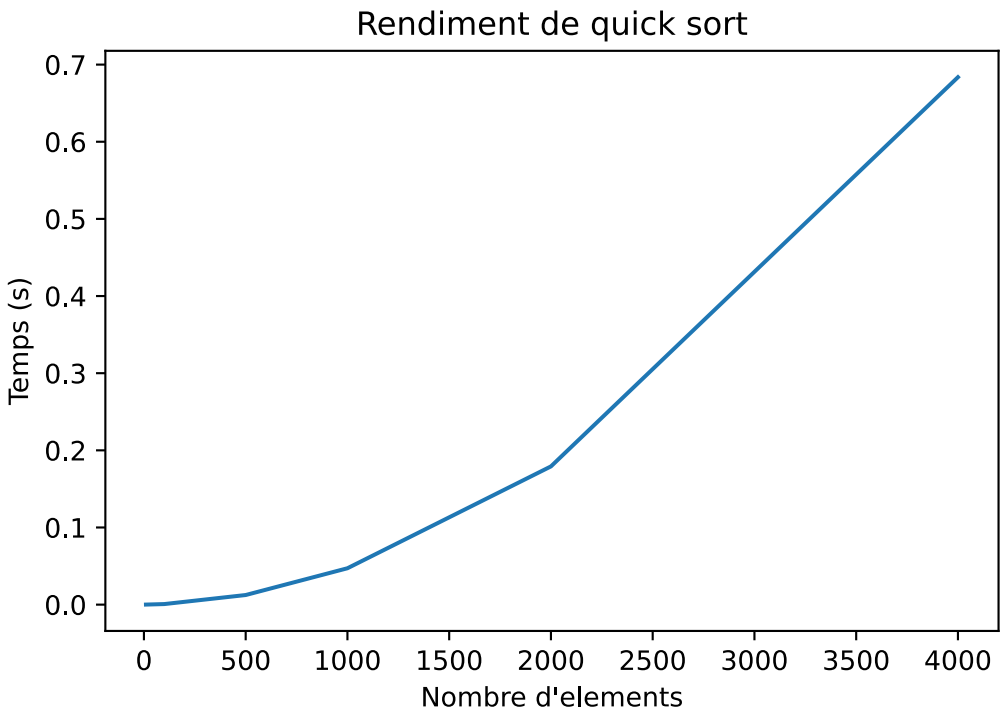
7.1. Com funciona?

7.2. Pseudocodi

7.3. Implementació

```python
#!/usr/bin/env python3
import utils

def sort(array):
    if len(array) < 2:
        return array

    pivot = array[0] # agafo el primer element com a pivot, pot ser qualsevol
    lower, higher = [], []

    for i in array[:-1]:
        if i < pivot:
            lower.append(i)
        else:
            higher.append(i)

    lower = sort(lower)
    higher = sort(higher)

    return lower + [pivot] + higher

if __name__ == "__main__":
    array = utils.numbers()
    print(sort(array))
```

7.4. Rendiment



8. Merge sort

8.1. Com funciona?

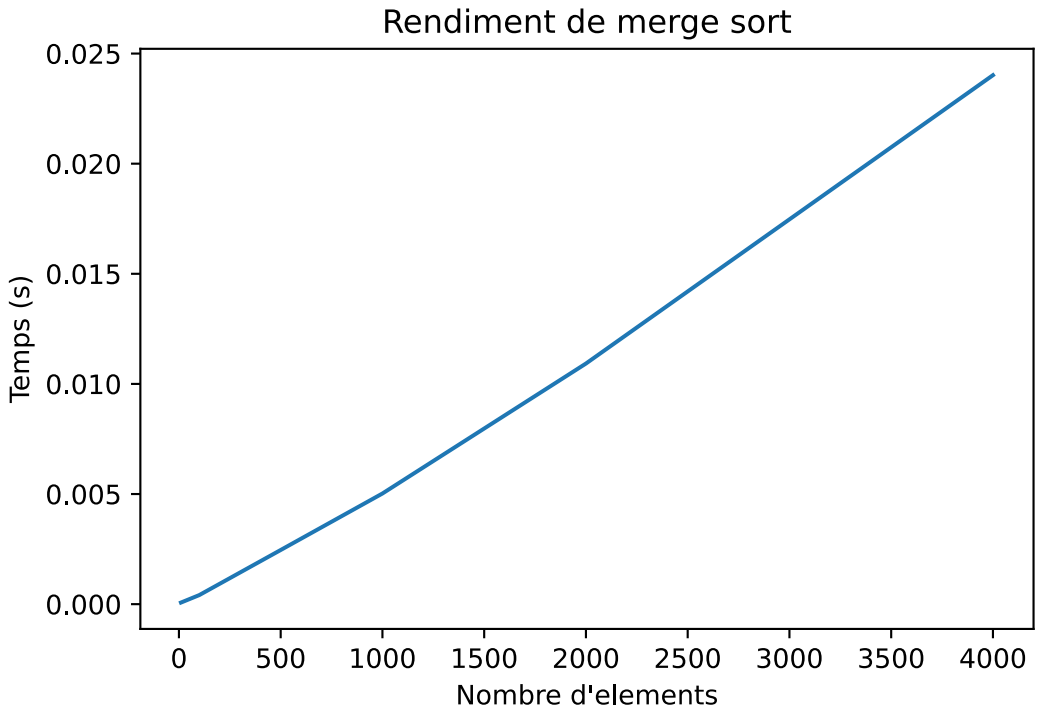
8.2. Pseudocodi

8.3. Implementació

```python
#!/usr/bin/env python3
import utils

def sort(array):
    if len(array) < 2:
        return array
    mid = len(array)//2
    left = sort(array[:mid])
    right = sort(array[mid:])

    merged = []
    l = 0
    r = 0

    while l < len(left) and r < len(right):
        if left[l] < right[r]:
            merged += [left[l]]
            l += 1
        else:
            merged += [right[r]]
            r += 1

    if l < len(left):
        merged += left[l:]
    if r < len(right):
        merged += right[r:]

    return merged

```

```
30 if __name__ == "__main__":
31     array = utils.numbers()
32     print(sort(array))
```

8.4. Rendiment



9. Comparació dels algoritmes