

# WORKSHEET 5: COMPUTATIONAL COMPLEXITY

Version 2.1  
Computing  
COMP110

Ed Powley

## Tasks

This worksheet tests your understanding of the concept of computational complexity, as well as your ability to communicate this understanding in writing.

Consider the following algorithm:

```
1: procedure HASDUPLICATE(list)
2:   let  $n$  be the length of list
3:   for  $i = 0, 1, \dots, n - 1$  do
4:     for  $j = 0, 1, \dots, n - 1$  do
5:       if  $i \neq j$  and  $\text{list}[i] = \text{list}[j]$  then
6:         return true
7:       end if
8:     end for
9:   end for
10:  return false
11: end procedure
```

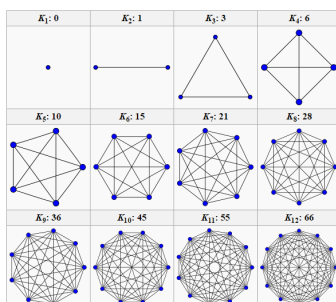
- (a) **State** what task this algorithm performs.
- (b) **Explain** why the worst case running time of the algorithm is quadratic, i.e.  $O(n^2)$ .

Now suppose that the **for** loop on line 4 is changed so that  $j$  ranges from 0 to  $i - 1$ .

- (c) **Explain** why the algorithm is still correct.
- (d) **Explain** why the algorithm will run approximately twice as fast.
- (e) Is the time complexity of the algorithm still quadratic? **Explain** your answer.

Now consider the following algorithm, which performs the same task as the first:

```
1: procedure HASDUPLICATE(list)
2:   let  $n$  be the length of list
3:   sortedList  $\leftarrow$  SORT(list)
4:   for  $i = 1, 2, \dots, n - 1$  do
5:     if sortedList[ $i - 1$ ] = list[ $i$ ] then
6:       return true
7:     end if
8:   end for
9:   return false
10: end procedure
```



Considering every pair of elements in a data structure often leads to quadratic complexity.

- (f) With reference to an appropriate source, **write down** the time complexity of Python's built-in `sort` function, in big- $O$  notation.
- (g) Thus **write down** the time complexity of the above algorithm in big- $O$  notation. **Explain** your answer.
- (h) If the size of the input list is large, which of these two algorithms is likely to run faster? **Explain** your answer.
- (i) **Suggest one** reason why a programmer might choose the "slower" algorithm over the "faster" one.

## Submission instructions

Begin by **forking** the GitHub repository at the following URL:

<https://github.com/Falmouth-Games-Academy/comp110-worksheet-5>

Write your answers to questions (a)–(i) in the `README.md` file. Open a **pull request**.

## Marking criteria

Remember that **it is better to submit incomplete work than to submit nothing at all**. Any attempt, even unfinished, will receive a passing grade.

Your work will be marked according to the following criteria:

- Where appropriate, are your answers **correct**?
- Are your explanations **clear, concise** and **accurate**?
- Where you have obtained information from external sources, are they **properly cited**?