Data Structures & Algorithms
Assignment 1 Briefing

Dr. Mark Matthews

Office: A1.13
School of Computer Science
University College Dublin

mark.matthews@ucd.ie

Learning Outcomes

On successful completion of this project the learner will be able to:

- Understand how to determine the amount of resources (such as time and storage) necessary to execute a particular algorithm (algorithm analysis).
- Understand the structure, nature and use of fundamental data structures
- Implement the data structures in Python.
- Understand how to use curve fitting (regression) approach to predict the equation of running time.
- Successfully write, compile, debug and run programs using these constructs.

Assignment 1 - Task 1

Factorial function:

1. Write an algorithm in pseudo-code to compute the factorial of a positive integer n

Assignment 1 - Task 2

Task 2: Factorial functions

1. Write 3 factorial functions to compute the factorial of a positive integer *n*. You should create 1 iterative factorial algorithm, 1 tail-recursive factorial algorithm and 1 non-tail recursive algorithm
2. Your algorithm should be able to take in any value of N and compute the factorial for that value
3. Execute each algorithms for a large range of input values of *n* and save the runtimes for each function call in a list, with the corresponding value of N.
4. For each of your algorithms, plot the running time as a function of *n* and compare the achieved results for each algorithm as well as to the complexity of the algorithm
5. *Note You may see that when you plot these results that you get spikes so you might want to run each multiple times and generate the average runtimes.

Assignment 1 - Task 3

Search Algorithms:

1. Use one of the sequences obtained from task 2 as an input for search algorithms.
2. Implement 1 linear search algorithm
3. Implement 1 binary search algorithm. You can choose to use either an iterative or a recursive search
4. For each approach, compute the best and worst running times. Give some examples by plotting the running time as a function of *n*
5. Execute each search algorithm multiple times with a range of search values and storing the performance times
6. *Note* for your binary search algorithm, you will first have to sort the sequence you are searching
7. Use curve-fitting algorithms to find the equation of the running times obtained from Tasks 1-3. See the attached

Your report should:

- Explain your methods and results including why the best and worst solutions performed the way they did and discuss any unusual results you find
- Have proper structure with introduction, main body covering the various areas you address, and conclusions
- Have satisfactory technical coverage, balancing breadth of coverage with depth
- Have soundness of argument
- Have good clarity of expression and level of readability
- Clearly reference sources of information, and avoid plagiarism

Note for Task 3, you will need to create several sequences of different sizes to assess the performance of your search algorithms.

Assessment: Things to consider

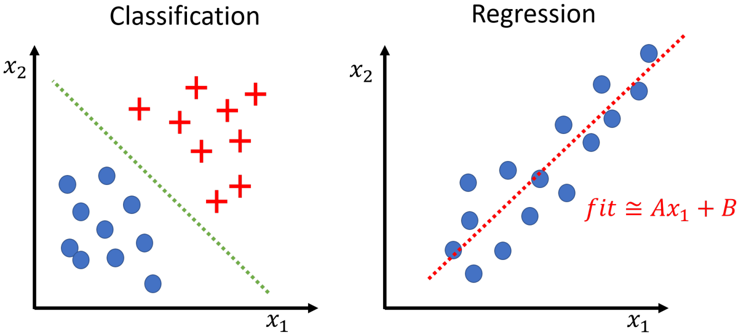
Speed: Test running time as input size grows. Compare running-time vs time complexity
Efficiency (operations/capacity): Can you estimate big-O (or big-theta), capacity requirements
Clarity: - are class names, method names, variable names meaningful; is a consistent style used throughout;
- code refactored (remove unnecessary or unused variables, loops etc.)
Correctness: - does it do what it is meant to do? - is it scaleable?
- is it hardcoded
Maintainability: - Are appropriate comments included in code?
- Are unit tests provided to test the key functionality

Assessment

Submission in week 7

- This assignment is worth 35% of the overall mark for COMP20230
- Your submission should be a single Python Notebook with code, analysis and graphs all contained within.
- It should be uploaded to the Brightspace by 5.00pm on Friday 31st March
- Submission: Please include your code, plots, and analyses as a single Notebook.

Example


Classification

Regression

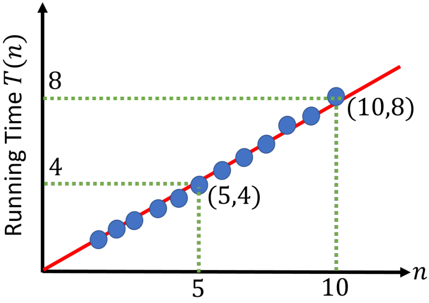
In the field of machine-learning, there are two common problems:

- Classification is to estimate a mapping function that predicts the class or category for a given observation
- Regression is to estimate a mapping function that predicts continuous output

Regression: Task 5

There are many algorithms to predict the fitting function, but let us take the simplest one as an example:

- The general form of linear running time is $T(n) = a * n + b$
- The problem is to find values of a (slope) and b (T-intercept)
- Use two points. Ex: $(n_1, T_1) = (5, 4)$ and $(n_2, T_2) = (10, 8)$
- $4 = 5 * a + b \dots$ (1) and $8 = 10 * a + b \dots$ (2)
- By solving the two equations, we get $a = \frac{4}{5}$ and $b = 0$
- $T(n) = \frac{4}{5}n$



Using Python?

A common Python library such as SciPy Curve Fitting can be used to predict the required fit function. More examples will be provided in Tutorial and Lab sessions.