Analysis and Design of Algorithms
Lecture 1

Introduction to Algorithms

Dr. Mohamed Loey

Lecturer, Faculty of Computers and Information
Benha University
Egypt

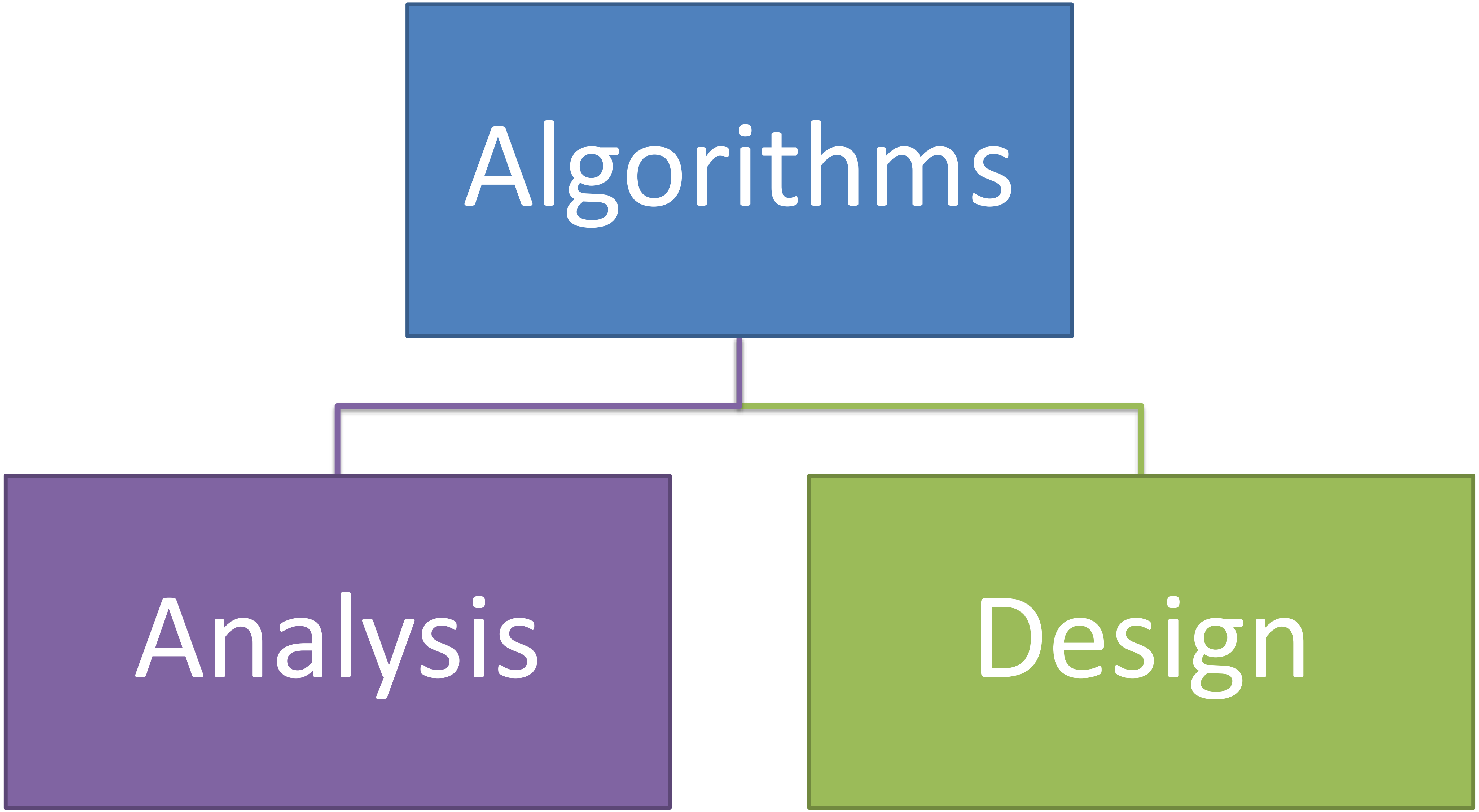
Table of Contents

- Algorithms
- Time Complexity & Space Complexity
- Algorithm vs Pseudocode
- Some Algorithm Types
- Programming Languages
- Python
- Anaconda

Algorithms

- An algorithm is a set of steps of operations to solve a problem performing calculation, data processing, and automated reasoning tasks.
- An algorithm is the best way to represent the solution of a particular problem in a very simple and efficient way.

Algorithms



Algorithms

Analysis

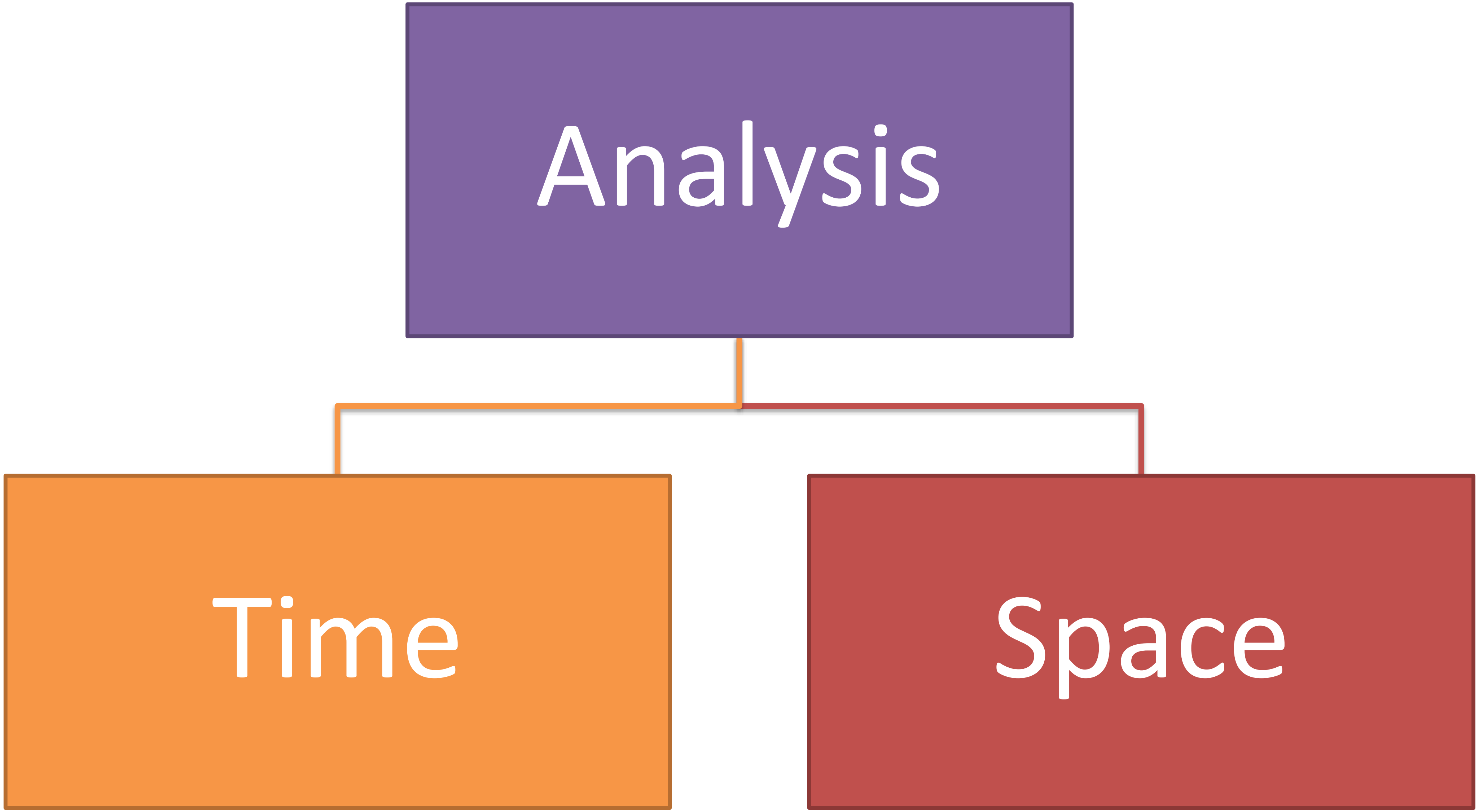
Design

Algorithms

- ❑ Analysis: predict the cost of an algorithm in terms of resources and performance

- ❑ Design: creating an efficient algorithm to solve a problem in an efficient way using minimum time and space.

Algorithms


Analysis

Time

Space

Time Complexity & Space Complexity

- Time Complexity is a function describing the amount of time required to run an algorithm in terms of the size of the input.

- Space Complexity is a function describing the amount of memory an algorithm takes in terms of the size of input to the algorithm.

Time Complexity & Space Complexity

- ❑ Time Complexity

What make algorithm **“fast”?**

- ❑ Space Complexity

How much **memory** is used?

Algorithms

□ Input: sequence $<a_1, a_2, \ldots, a_n>$ of numbers.

□ Output: permutation $<a'_1, a'_2, \ldots, a'_n>$ such that

$$a'_1 \leq a'_2 \leq \ldots \leq a'_n \ .$$

Example:

Input	8	12	5	9	2
Output	2	5	8	9	12

Algorithm vs Pseudocode

- An algorithm is a formal definition with some specific characteristics that describes a process. Generally, the word "algorithm" can be used to describe any high level task in computer science.

- Pseudocode is an informal and human readable description of an algorithm leaving many details of it. Writing a pseudocode has no restriction of styles and its only objective is to describe the high level steps of algorithm.

Algorithm vs Pseudocode

- ❑ **Algorithm: Selection Sort**

  Input: A list L of integers of length n

  Output: A sorted list L1 containing those integers present in L

  Step1: Find the minimum value in the list L

  Step2: Swap it with the value in the current position

  Step3: Repeat this process for all the elements until the entire list is sorted

  Step 4: Return the sorted list L1

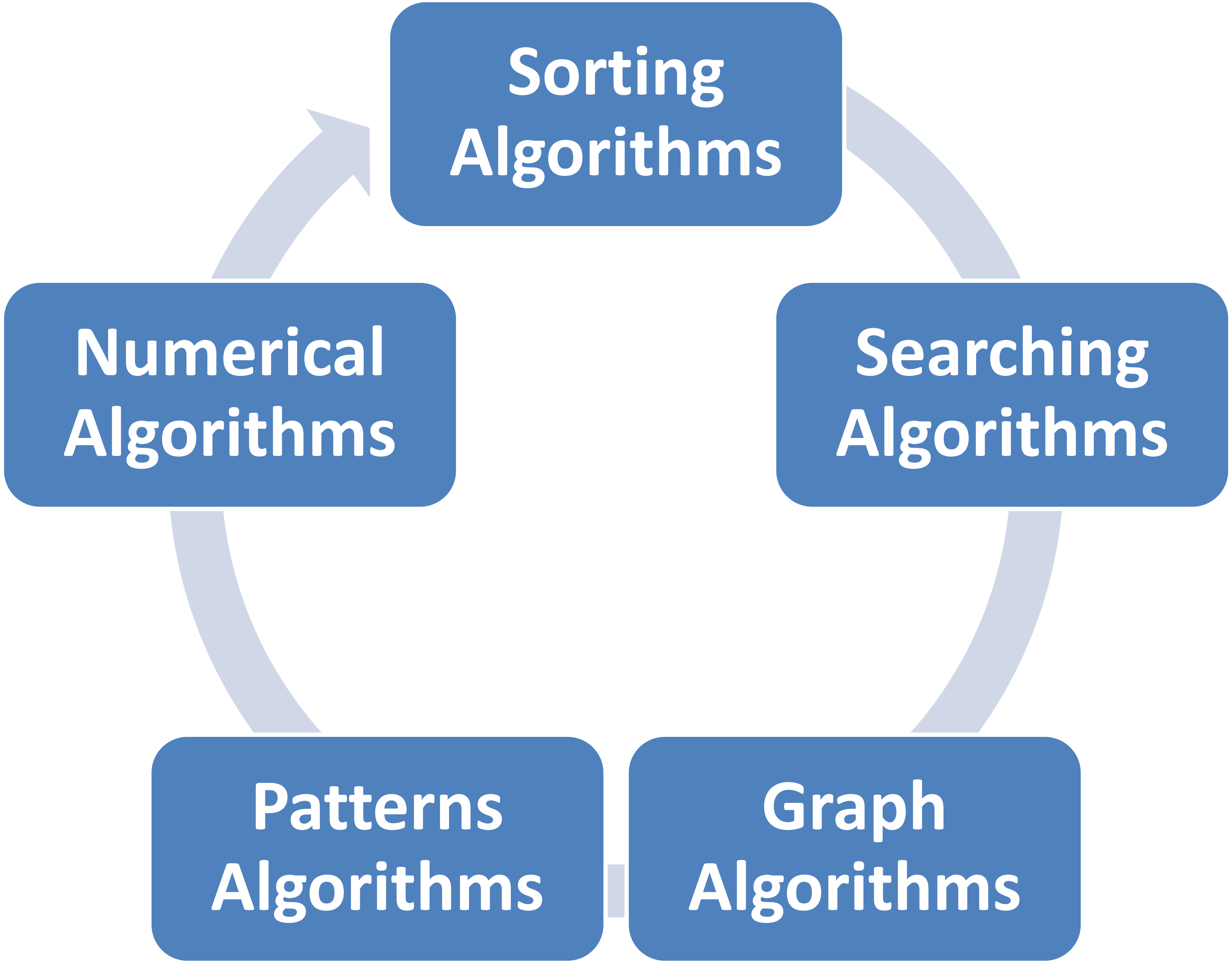
  Step 5: Stop

Algorithm vs Pseudocode

❑ Pseudocode : Selection Sort

```
for j ← 1 to n-1
    smallest ← j
    for i ← j + 1 to n
        if A[ i ] < A[ smallest ]
            smallest ← i
    Exchange A[ j ] ↔ A[ smallest ]
```

Some Algorithm Types



Sorting Algorithms

Searching Algorithms

Graph Algorithms

Patterns Algorithms

Numerical Algorithms

Some Algorithm Types

- ❑ **Sorting Algorithms** are to rearrange the items of a given list in non decreasing order.

- ❑ **Searching Algorithms** deal with finding a given value, called a search key, in a given set.

Some Algorithm Types

- ❑ **Pattern (String) Algorithms** deal with string which comprise letters, numbers, and special characters; bit strings, which comprise zeros and ones; and gene sequences

- ❑ **Numerical Algorithms** deal with mathematical problems that solving equations and systems of equations, computing definite integrals ,evaluating functions, and so on.

Some Algorithm Types

- ❑ **Graph Algorithms** deal with graphs. Graph can be thought of as a collection of points called vertices, some of which are connected by line segments called edges. Graphs can be used for modeling a wide variety of applications, including transportation, communication, social and economic networks, project scheduling, and games.

Programming Languages

- ❑ All algorithms can be coded using any programming language such as (Pyhton, C++, Java, PHP, JavaScript, C#, …)

- ❑ The most used programming language in this course is Python

- ❑ Why Python???

Python

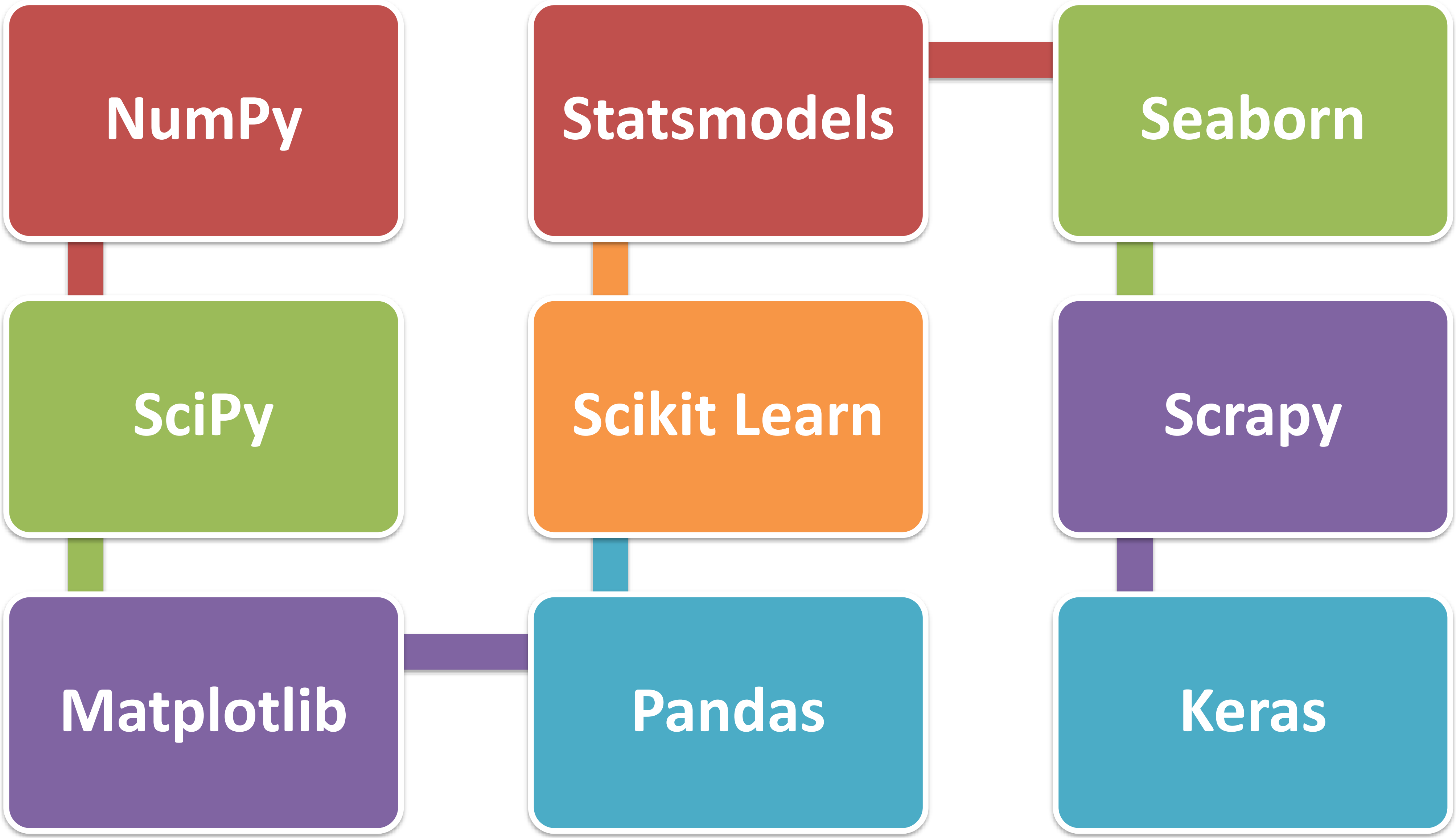
1) **Easy to Understand:**

- ❖ Python is very high level language, Python reads like English. Python is incredibly easy to learn and use.

2) **Python Has Amazing Libraries**

- ❖ When you're working on bigger projects, libraries can really help you save time and cut down on the initial development cycle.

Python



NumPy

SciPy

Matplotlib

Pandas

Scikit Learn

Statsmodels

Seaborn

Scrapy

Keras

Python

3) Supportive community

- ❖ Python has documentation, guides, tutorials and more. Plus, the developer community is incredibly active.

4) Great Corporate Sponsor

- ❖ C# has Microsoft, Java has Sun and PHP is used by Facebook. **Google** adopted Python heavily back in 2006, and they've used it for many platforms and applications since.

Python

5) Python can do:

Python can do

Desktop apps & Web apps

Data mining

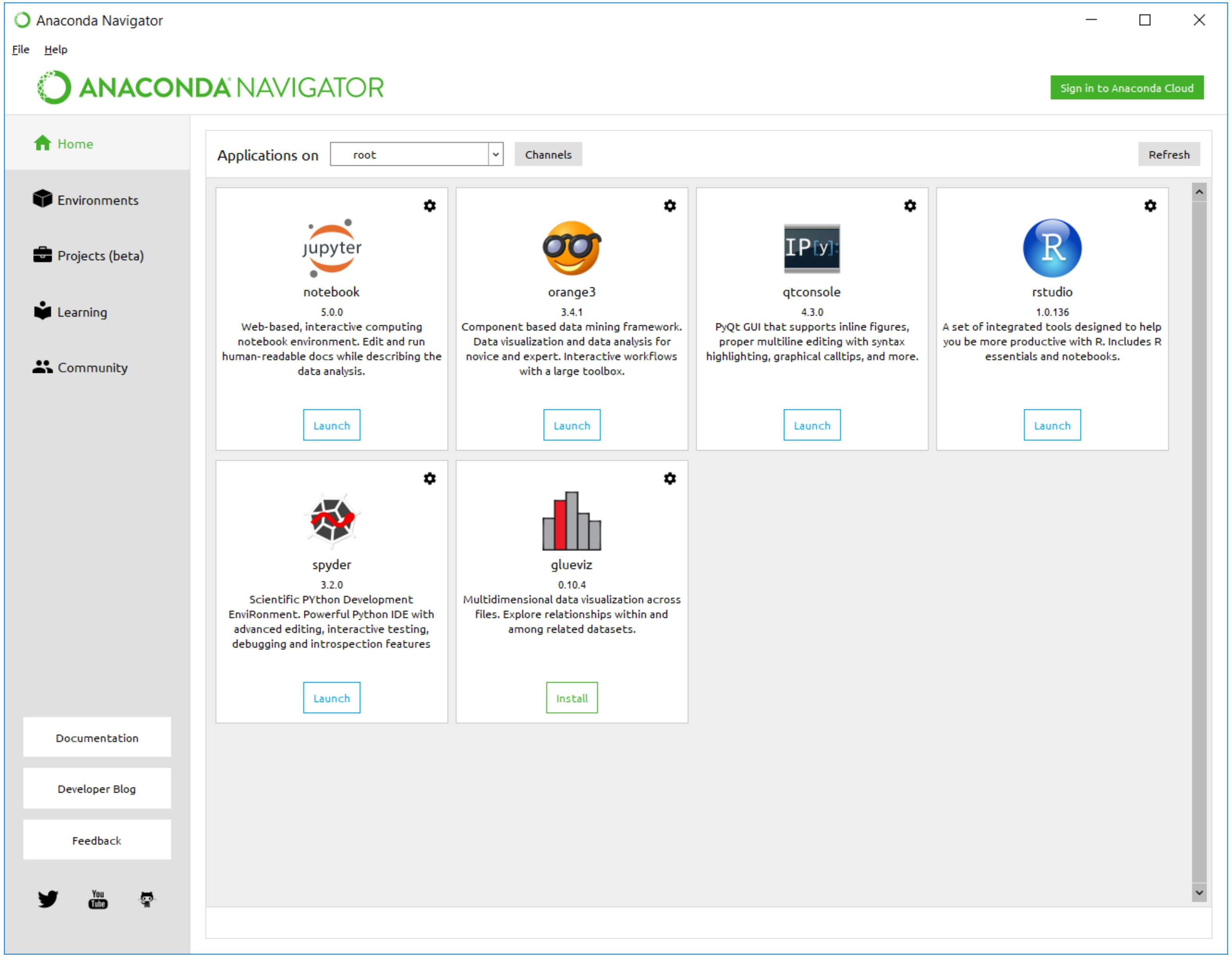
Scientific computing

Python

6) Python distribution:

- The best Python distribution, we will used in our course is **Anaconda**

Anaconda



notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

orange3
3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

qtconsole
4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

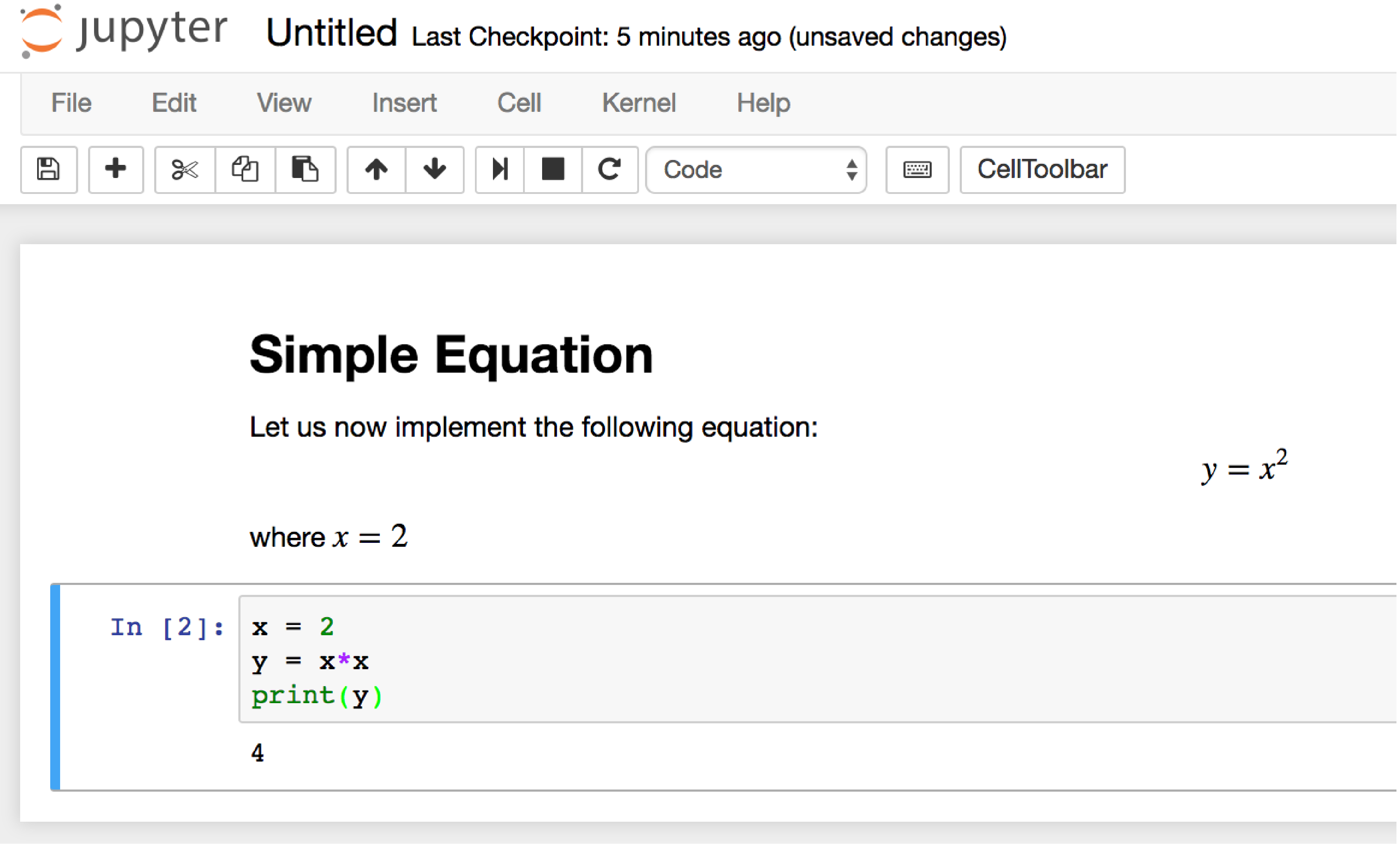
rstudio
1.0.136
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

spyder
3.2.0
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

glueviz
0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Anaconda

- ❑ Jupyter notebook



Untitled Last Checkpoint: 5 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help

Simple Equation

Let us now implement the following equation:

$$y = x^2$$

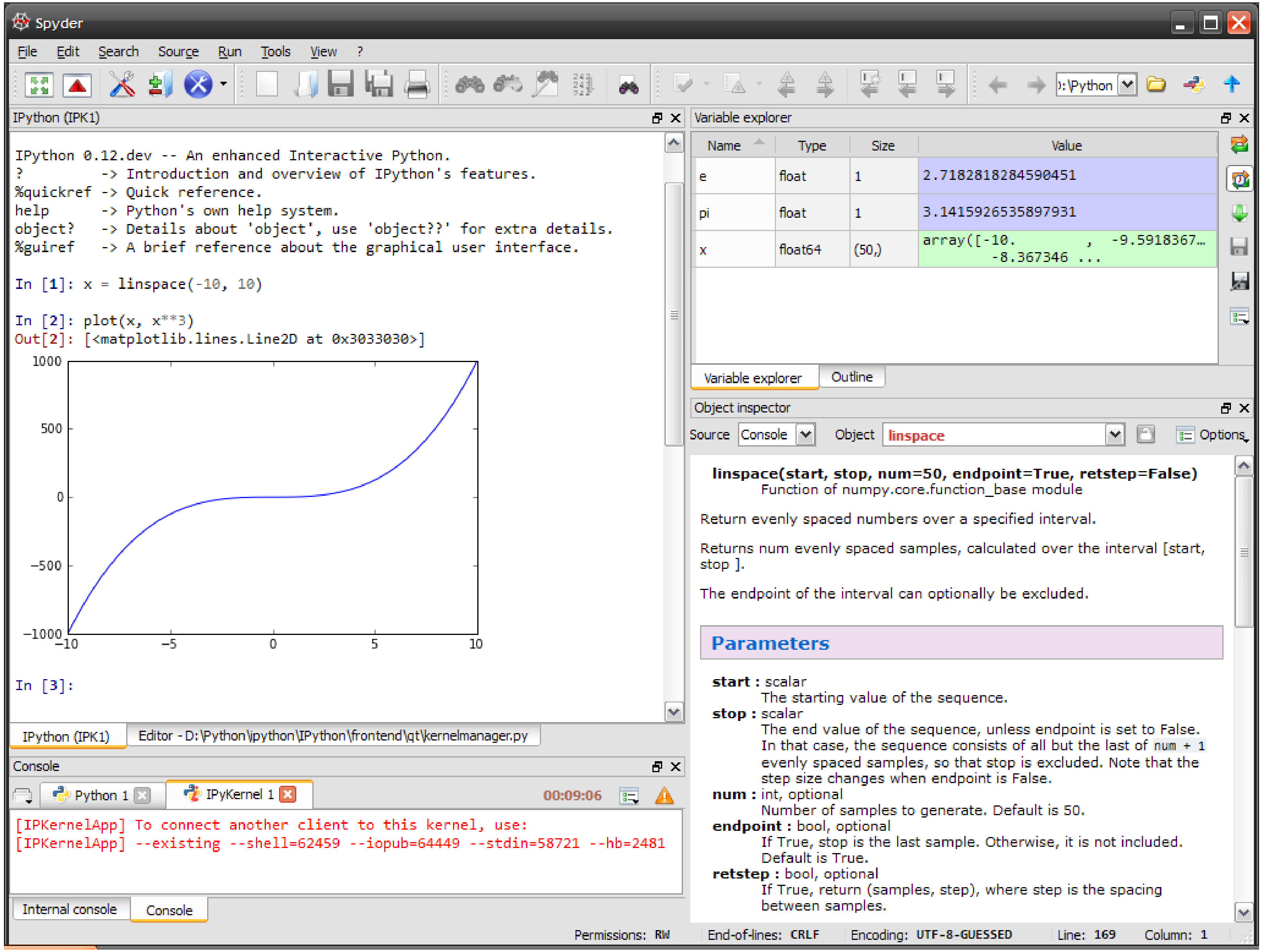
where $x = 2$

In [2]:
```
x = 2
y = x*x
print(y)
```

```
4
```

Anaconda

- Spyder



```
IPython 0.12.dev -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%guiref   -> A brief reference about the graphical user interface.

In [1]: x = linspace(-10, 10)

In [2]: plot(x, x**3)
Out[2]: [<matplotlib.lines.Line2D at 0x3033030>]

In [3]:
```

Name	Type	Size	Value
e	float	1	2.7182818284590451
pi	float	1	3.1415926535897931
x	float64	(50,)	array([-10. , -9.5918367… -8.367346 ...

linspace(start, stop, num=50, endpoint=True, retstep=False)
Function of numpy.core.function_base module

Return evenly spaced numbers over a specified interval.

Returns num evenly spaced samples, calculated over the interval [start, stop].

The endpoint of the interval can optionally be excluded.

Parameters

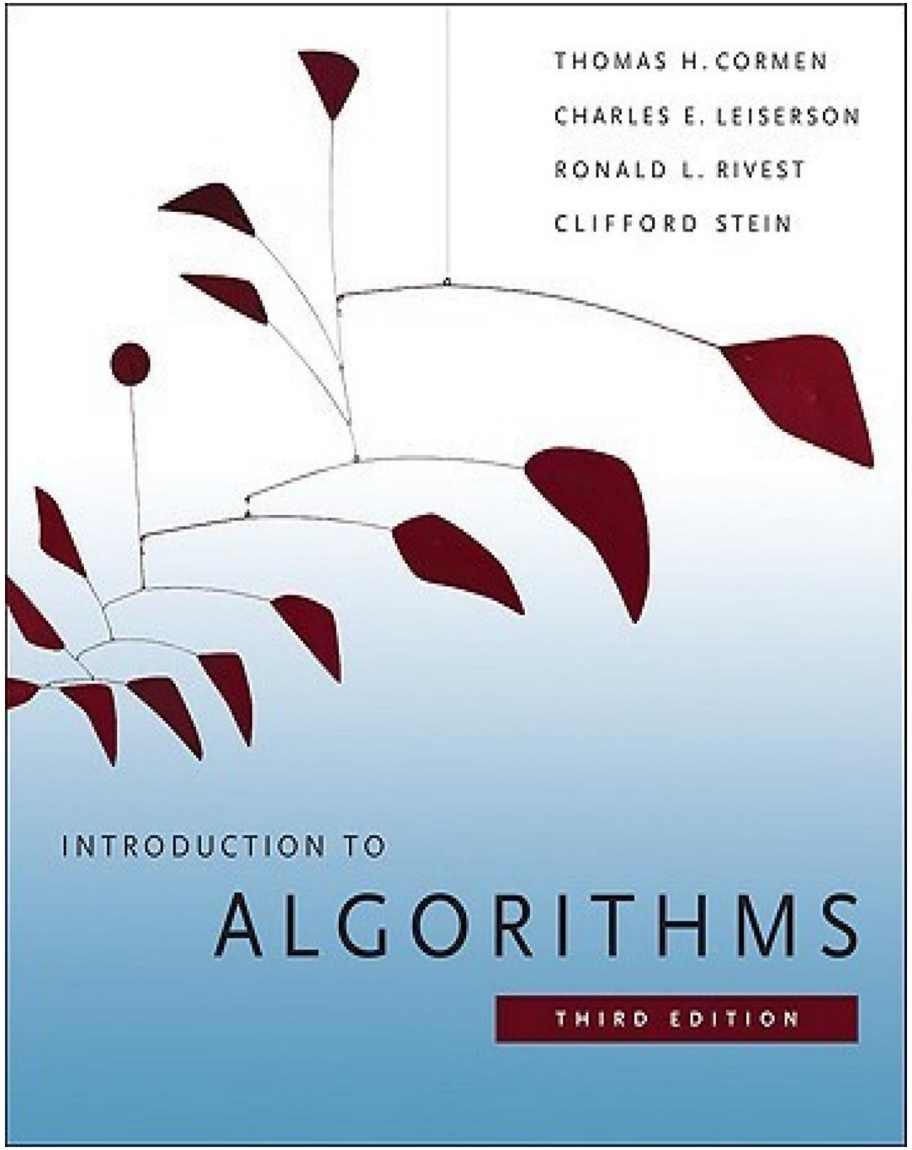
start : scalar
The starting value of the sequence.
stop : scalar
The end value of the sequence, unless endpoint is set to False. In that case, the sequence consists of all but the last of num + 1 evenly spaced samples, so that stop is excluded. Note that the step size changes when endpoint is False.
num : int, optional
Number of samples to generate. Default is 50.
endpoint : bool, optional
If True, stop is the last sample. Otherwise, it is not included. Default is True.
retstep : bool, optional
If True, return (samples, step), where step is the spacing between samples.

```
[IPKernelApp] To connect another client to this kernel, use:
[IPKernelApp] --existing --shell=62459 --iopub=64449 --stdin=58721 --hb=2481
```

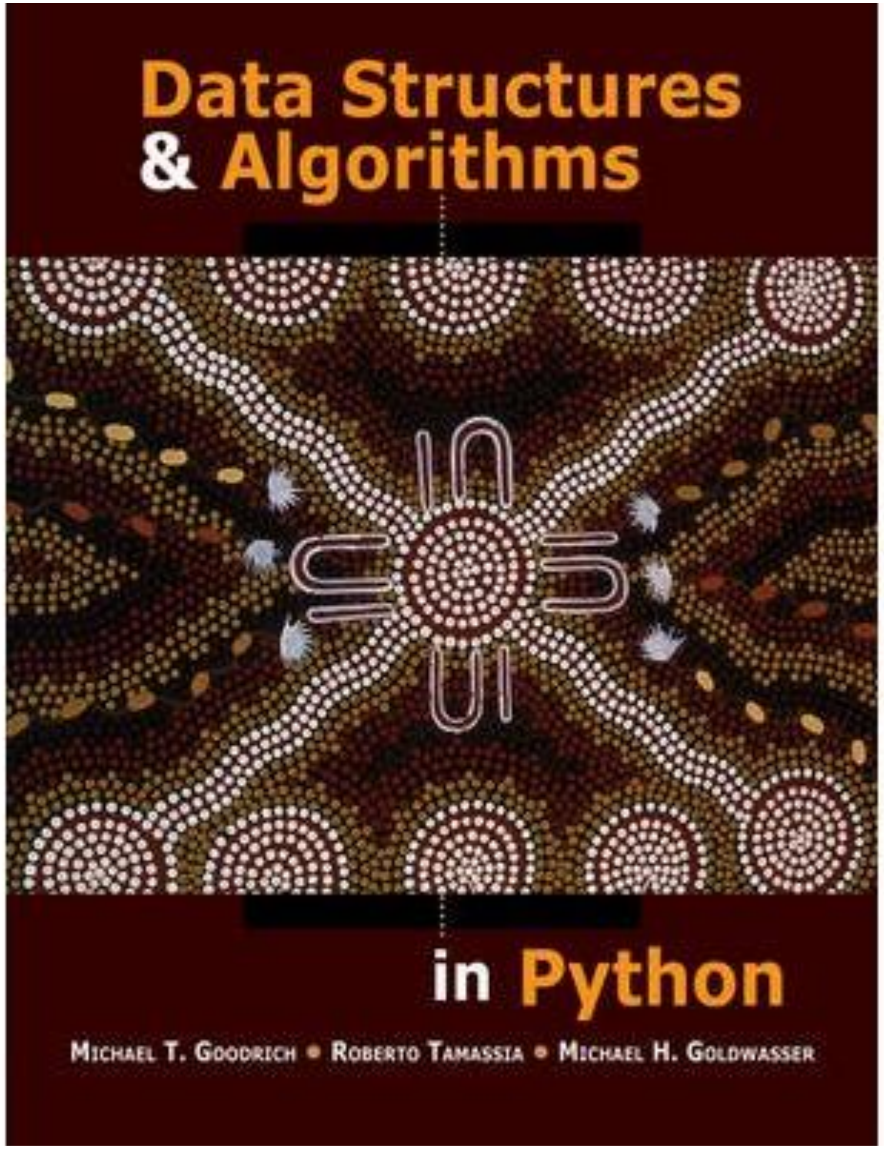
Permissions: RW   End-of-lines: CRLF   Encoding: UTF-8-GUESSED   Line: 169   Column: 1

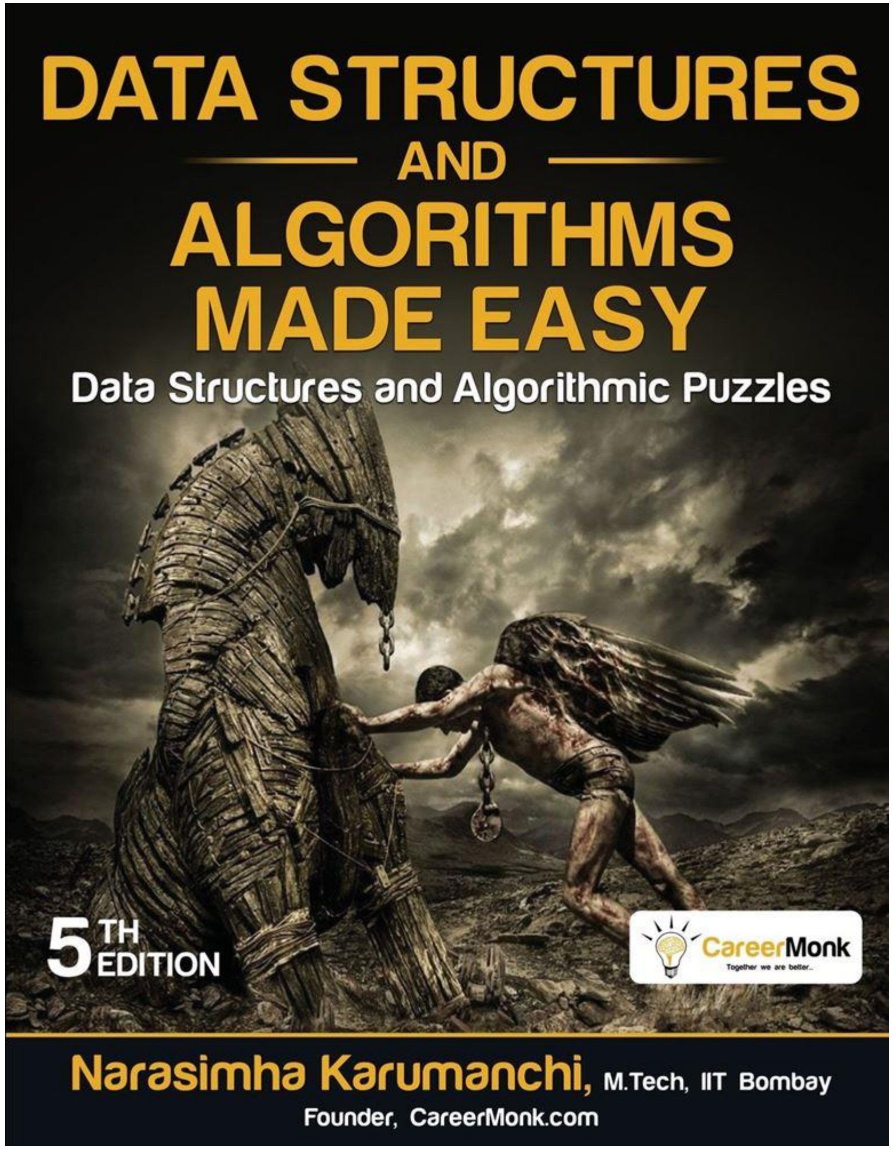
Books

Algorithms
FOURTH EDITION
ROBERT SEDGEWICK | KEVIN WAYNE

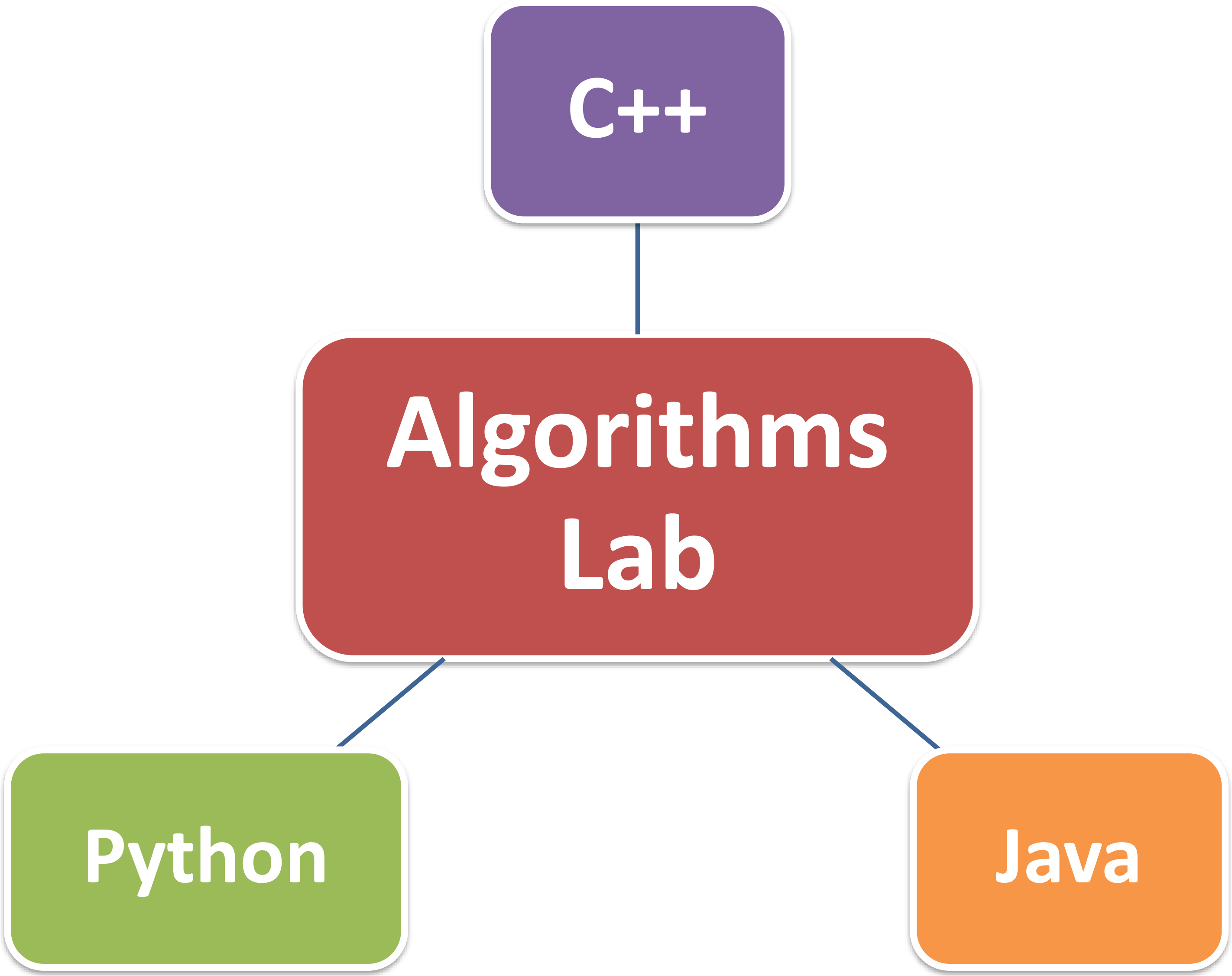

THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN
INTRODUCTION TO
ALGORITHMS
THIRD EDITION

Books


Data Structures
& Algorithms
in Python
Michael T. Goodrich • Roberto Tamassia • Michael H. Goldwasser


DATA STRUCTURES
AND
ALGORITHMS
MADE EASY
Data Structures and Algorithmic Puzzles
5TH EDITION
CareerMonk
Together we are better..
Narasimha Karumanchi, M.Tech, IIT Bombay
Founder, CareerMonk.com

Algorithms Lab



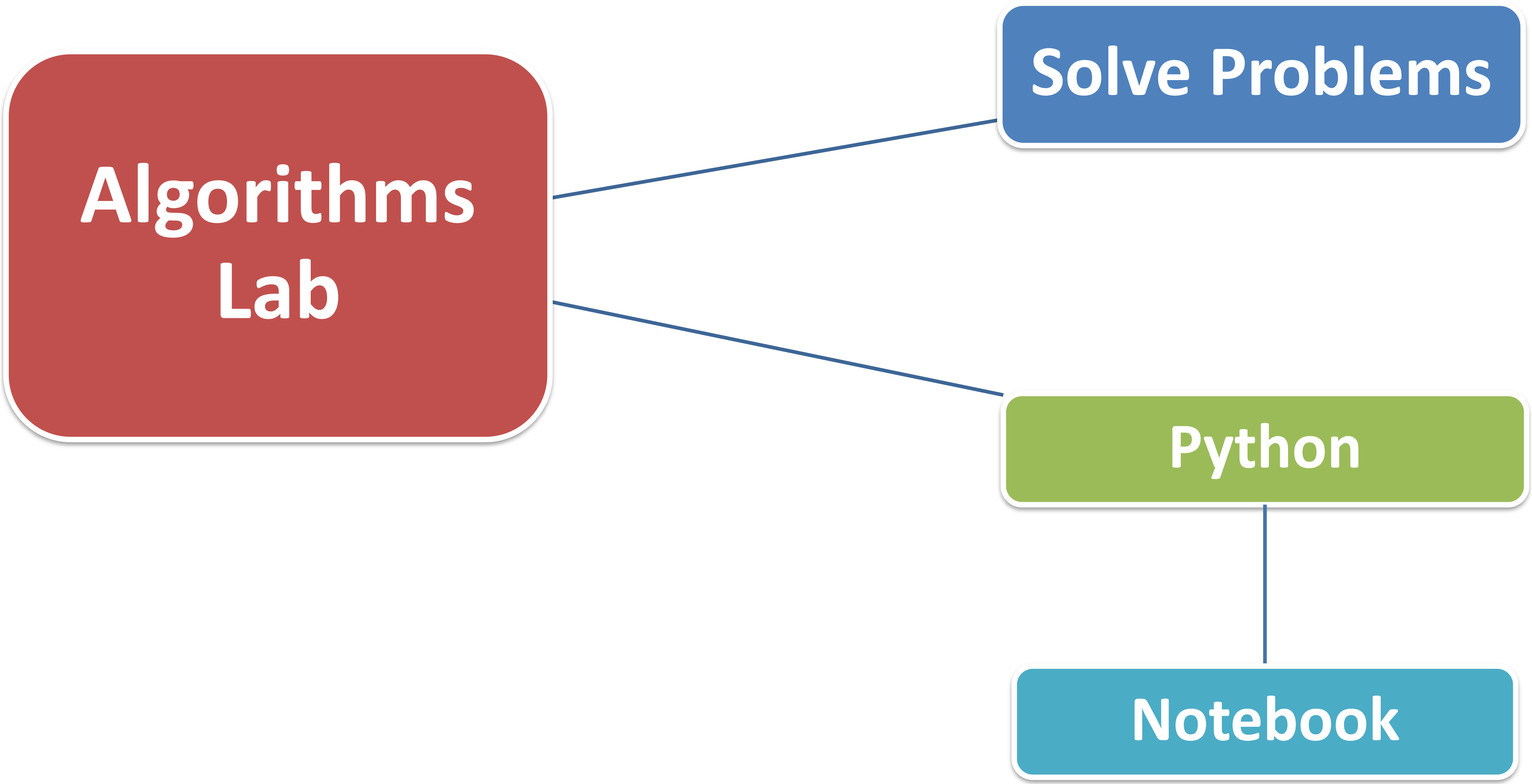
C++

Algorithms Lab

Python

Java

Algorithms Lab



Contact Me

facebook.com/mloey

mohamedloey@gmail.com

twitter.com/mloey

linkedin.com/in/mloey

mloey@fci.bu.edu.eg

mloey.github.io

THANKS FOR YOUR TIME