# AIMA Code Installation Instructions

Implementations of all algorithms presented in the lecture in several programming languages are available online at `https://github.com/aimacode`. For most of the examples from the lecture we provide *Jupyter Notebooks* that implement the example in Moodle. This should encourage you to debug the code for the examples step by step in order to develop a better understanding of the involved algorithms. The following two steps are required to set up a programming environment that allows you to execute the *Jupyter Notebooks*:

1. Installation of *Anaconda*
2. Download of the AIMA python code

## 1 Installation of Anaconda

To execute the *Jupyter Notebooks* it is required to first install *Python*, *Jupyter*, and several standard python libraries. We recommend to use the *Anaconda* environment, which installs all the above mentioned programs at once including the package manager conda. Conda is also used to create a virtual environment.

*If you already use conda or want to use the python environment 'venv', or simply your home python distribution, feel free to do so and jump directly to point 2*

### 1.1 Installation on Linux

1. Download the Python 3 (currently 3.7) installer from:

   `https://www.anaconda.com/download/#linux`

2. Go to the download folder your terminal and run:

   ```
   bash Anaconda-latest-Linux-x86_64.sh
   ```

3. Follow the prompts on the installer screens. If you are unsure about any setting, accept the defaults. You can change them later. One of them is the auto 'conda init'. It will initialize the conda base environment each time you start your terminal.

4. To make the changes take effect, close and then re-open your terminal.

5. To test your installation, in your terminal or Anaconda prompt, run the following command to list all installed packages:

   ```
   conda list
   ```

### 1.2 Installation on Windows

1. Download the Python 3 (currently 3.7) installer from:

   `https://www.anaconda.com/download/#windows`

2. Double-click on the *.exe* file.

3. Follow the instructions on the screen. If you are unsure about any setting, accept the defaults. You can change them later.

4. When installation is finished, form the *start* menu, open the *Anaconda prompt*.

5. To test your installation, in your anaconda prompt, run the following command to list all installed packages:

   ```
   conda list
   ```

### 1.3 Installation on macOS

1. Download the Python 3 (currently 3.7) installer from:

   `https://www.anaconda.com/download/#macos`

2. Double-click the *.pkg* file.

3. Follow the prompts on the installer screens. If you are unsure about any setting, accept the defaults. You can change them later.

4. To make the changes take effect, close and then re-open your terminal.

5. To test your installation, in your terminal or anaconda prompt, run the following command to list all installed packages:

   ```
   conda list
   ```

### 1.4 How to use Anaconda

Anaconda distribution comes with more than 1,500 packages as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). First time users might find helpful information in the anaconda docs:

`https://docs.anaconda.com/anaconda/navigator/`

As you will see in the following section we will use an Anaconda environment for package managing. An introduction to how to use Anaconda within the command line can be found here:

`https://conda.io/projects/conda/en/latest/user-guide/getting-started.html`

The most important commands are:

1. Creating a new Anaconda environment:

   ```
   conda create --name <env_name>
   ```

2. List all existing environments:

   ```
   conda info --envs
   ```

3. Activate specific environment:

   ```
   conda activate <env_name>
   ```

4. Install package:

   ```
   conda install <package_name>
   ```

5. List all packages of current environment:

   ```
   conda list
   ```

## 2 Download of the AIMA python code

Python implementations for the algorithms from the lecture are available on the repository at `https://github.com/aimacode/aima-python`. For installation, the following steps are required:

1. If the application *git* is not yet installed on your machine or in your *Anaconda* application install it with the following command

   ```
   conda install -c anaconda git
   ```

2. Download the repository[1]

   ```
   git clone https://github.com/aimacode/aima-python.git
   ```

3. Create a new Anaconda environment. Here it is assumed that the environment is called AI_AIMA.

   ```
   conda create --name AI_AIMA python=3.7
   ```

   (This step is not required. You can also work within the base environment. Nevertheless using environments makes it easier to distribute your projects later on.)

   If not yet activated, activate your environment. This step is needed each time you want to work within the environment. The current environment is indicated left to your computers name in the terminal.

   ```
   conda activate AI_AIMA
   ```

4. Install pip within your conda environment:

   ```
   conda install pip
   ```

5. Go inside the project folder and install the project requirements:

   ```
   cd aima-python
   pip install -r requirements.txt --use-feature=2020-resolver
   ```

[1] Tested with commit 668a2fb0bcd28b4963648c1425f904baa3826a8f

This will fetch all python packages needed. Unfortunately *conda* has some issues installing *opencv* so we used *pip* in this case. Usually it is easier to just use *'conda install'* to install the needed packages.

The `--use-feature=2020-resolver` may be necessary if your pip installation is not the latest version.

The download may take a while. It case it fails to download some of the packages (for example due to connection problems), try to run the command again.

6. Check if the packages are installed:

   ```
   conda list
   ```

7. Fetch the corresponding dataset from the *aima-data* repository:

   ```
   git submodule init
   git submodule update
   ```

   The download of the set may take a while.

8. Run the test suite:

   ```
   py.test
   ```

   If all tests were successful, you are now ready to start!

   If not, look below in the known bugs/FAQ section.

## 3 Executing the Jupyter notebooks

For most of the examples from the lecture we provide *Jupyter Notebooks* on Moodle. To avoid issues with the relative file path we recommend to place these notebooks in the root folder of the *AIMA* repository you downloaded in the previous step. To start the *Jupyter* web-interface simply type the following command into your terminal / anaconda prompt:

```
jupyter notebook
```

From the web-interface you can then easily open, modify, and execute the *Jupyter Notebooks*. Depending on your environment, it is possible that you have to install some additional python libraries. This can be done with the command:

(Note: Make sure you have activated your **project environment** *'AI_AIMA'*.)

```
pip install <library name>
```

or

```
conda install <library name>
```

# 4 Known Bugs/FAQ

## 4.1 When executing `py.test`, the following tests fail: `test_deep_learning4e.py`, `test_learning4e.py`, and `test_perception4e.py`

This is a known bug investigated here:
`https://github.com/aimacode/aima-python/issues/1231`
While waiting for a fix, the following workaround can be used instead: From a ***different*** console than the one you use to setup AIMA, run the following lines:

```
pip install pypi-timemachine
pypi-timemachine 2020-10-10
```

This will return a result looking as follows:

```
Starting pypi-timemachine server at <YOUR LOCAL ADDRESS>
```

where `<YOUR LOCAL ADDRESS>` is an address that is specific to your machine. Then, from the ***original*** console you used for the AIMA setup, run the following:

```
pip install --index-url <YOUR LOCAL ADDRESS> -r requirements.txt --force-reinstall
```

You can then run `py.test` again.

## 4.2 In Step 5 of installing the AIMA package, the `pip install` command returns an error, when installing `qpsolvers` or `quadprog`.

**For Windows:** You may need the latest version of the C++ compiler provided by Visual Studio. You can download it here:
`https://visualstudio.microsoft.com/visual-cpp-build-tools/`
After installation, reboot your system. This should fix the problem.

**For iOS:**
This is a known bug related to an outdated version of Xcode. You can update Xcode by running the following command:

```
xcode-select --install
```

You can find more information here:
`https://stackoverflow.com/questions/58364832/problems-installing-qpsolvers-on-mac`