

# 1729043774-prelab

## Pre-Lab

### The purpose of the Pre-Lab

The Pregel Lab is a hands-on activity that will help you understand how the Pregel model works. You'll write some Pregel algorithms in Go, using the provided Go-Pregel framework. However, to go through the lab, you will need to setup your environment. Although setting the environment is not hard (it is basically cloning the repo and installing some dependencies), it can be a little time-consuming since some dependencies take time to download. That's why we have the Pre-Lab: to help you set up your environment at home, so you can focus on the lab itself during the class.

### The Pre-Lab

1. Go to <https://github.com/GaGandour/Go-Pregel> and clone the repo. You can do this by clicking on the green "Code" button and copying the URL, then running `git clone <URL>` in your terminal. You can also download the zip file and extract it on your machine.
2. Ensure you have Docker installed on your machine. If you don't have it, you can download it at <https://www.docker.com/>.
3. (Optional, but recommended) If you don't have Go installed on your machine, you can download it at <https://go.dev/doc/install>.
4. If you are using Windows, you'll need to go through this [small section on the README.md file](#).
5. Similarly, if you are using Linux, you'll need to go through this [small section on the README.md file](#).
6. Create a python virtual environment. You can follow the instructions on the [section about the python environment](#)
7. Generate the missing files. Some files are missing from the repository, and Go-Pregel won't run unless you create them. You can create them by running the following command in the project directory:

```
# Run from the root of the project
cd scripts/prepare-repo
./write_untracked_files.sh
cd ../..
```

## Making sure everything is working

After you've done all the steps above, you can run the following command to check if everything is running smoothly:

```
# Run from the root of the project
cd scripts/execution
./start_pregel.sh -num_workers=1 -graph_file=sssp/graph1.json
cd ../..
```

You should see, in your browser, a graph image with some vertices and edges. If you see that, your python environment is working properly. Check your terminal to see if there are any errors on the docker container. If your computer started the docker container and stopped it with no errors, you are ready for the lab.