What You Will Need

1. The python package installer "pip". (Note that you might need pip3.) See this link: https://pip.pypa.io/en/stable/installing/
2. Graphviz, a graph visualization software. See this link: https://graphviz.org/
3. A basic knowledge of the DOT language that Graphviz uses to create graph visualizations (it's not hard I promise). I will include some basic examples that you can use to get you started. Documentation for this language can be found on the Graphviz website. However, better help can almost always be found on stackexchange.
4. After you have pip or pip3 installed, you will need to install the following python packages
   (a) NetworkX (See here https://networkx.org/documentation/stable/install.html for installation instructions),
   (b) PyGraphviz (See here https://pygraphviz.github.io/documentation/stable/install.html for installation instructions).

Using assignment graph generator

Once you have the above downloaded, you will need to create the base assignment graph in the DOT language. I will provide you with some base graphs to start with (e.g. C_3, C_4, the Peterson Graph, etc.)

If you want to create your own base graphs, I would suggest copying the base graphs you have already created using the cp command. For example:

```
cp C4.dot new_base_graph.dot
```

Then open your new base graph dot file in your favorite text editor, adding and subtracting edges as you wish.

Once you have a base graph to feed in as input to the assignment graph generator, then input the following code into the command line:

```
python3 sg_brack_generator.py

 Input dot file name: name_of_your_base_graph.dot

 dot file for assignment graph: name_of_your_output_file.dot
```

You will see a slew of output that ends with numbers separated by lines. These numbers at the end are the length of the cycles in the cycle basis of the assignment graph. Currently the program only prints the nodes in the cycle if the cycle is of length 6 or 8.

The name of your output file is the file to which the program will write the DOT encoding of the assignment graph. See the visualizing section for directions on how to visualize this using Graphviz.

Note: The program currently only gives a $(2, 1, 1, \ldots, 1)$ pebbling assignment to the base graph. However, this can be easily changed.

Visualizing Graphs Using Graphviz

Now that the program produced a dot file encoding of The best way to visualize **assignment graphs** is using the Graphviz command line tool. For example to visualize the file

```
peterson_brackets.dot
```

type at the command prompt

```
dot -Tpng peterson_brackets.dot -o peterson_brackets.png
```

This will output the graph visualization to the file specified after the -o flag.

To visualize base graphs (especially those that have cycles), use the following command at the commandline

```
sfdp - Tpng base_graph.dot -o base_graph.png
```

The "dot" graph visualization algorithm that comes with the Graphviz installation is better for graphs with a hierarchical structure. Thus it is perfect for assignment graph visualization but not for visualizing graphs that do not have a hierarchical structure such as most of the base graphs we have been studying. Indeed, the base graph visualization using the "dot" command will look very strange and unhelpful.