Workshop session 8

This workshop will be about algorithm analysis and big-O notation.

1. Go through the slides to make sure you understand

- Constant, linear, quadratic, cubic, exponential and logarithmic growth
- How to measure the running time of a program
- Big-O notation

2. The file `turtle_fractal.py` contains a program that draws a fractal.

- Unfortunately no one really understands how it works. Help us by going through the program line by line (possibly in the debugger) and make sure you understand what each line is doing. If you'd like, you can add comments that describe the code.
- Next, call the variables something more meaningful. The names `t` , `lv` and `s` does not make any sense.
- Now time how long it takes for the turtle to draw the whole shape.
- Increase the previous `lv` variable to 6. Measure the runtime.
- Increase the previous `lv` variable to 7. Measure the runtime.
- How does the runtime scale with the `lv` variable? What is the worst-case big-O runtime?
  - **Hint**: Try to plot the runtimes you have for `lv` on a plot, where teh x-axis is the `lv` value and the y-axis is the runtimes.

3. Take the file `befkbhalderstatkode_small.csv` and write a small program that:

- Read each line of the file except the very first line (the header line), which contains the string `AAR,BYDEL,ALDER,STATKODE,PERSONER` .
- Read the values of each column into a separate list. They could be called `aar` , `bydel` , `alder` , `statkode` , and `personer` respectively.
- Compute the total amount of people that lived in Copenhagen in 2015. The amount of people is stored in the column `PERSONER`` . Print the result.
- **Hints**: remember how to read files all of the following keywords (each bullet point corresponds to a line in your program)
  - `with` , `open` , `filename`
  - Skipping a line? Check today's recap slides.
  - `loop over file`
  - splitting a string on a certain character
  - appending elements to a list
  - You can compute the sum of all integers in a list using the `sum` function (read `print(help(sum))`) or try to run and understand:

```
values = [1, 2, 3, 5, 8]
result = sum(values)
print(result)
```

4. Write a program that reads each line of a file and that prints each line preceded by a line number.

- That is, your program should print

```
1. MOVE 10
2. TURN 90
3. MOVE 10
4. TURN 90
```

for a file containing the following (actually one of the earlier turtle scripts):

```
MOVE 10
TURN 90
MOVE 10
TURN 90
```

- **Hints**:
  - Remember how you can use the `enumerate` function to get the index number and the actual element of a list in a `for` -loop
  - Remember how to destructer (unpack) multiple values
  - In essence, check the recap slides of session 7 or the first slides in `03_Big_O_Notation.ipynb` .