CENG 218 – Analysis and Design of Algorithms
Homework 1

8 April 2020

Due Date: 20 April 2020

Exercise 1 Tracking Covid-19 (70/100 pts)

- Download and extract the contents of `ceng218_hw01.tar.gz`.
- We will read, sort, and display the data from the CSV file `ceng218_hw01/data/covid_data_full_2020_04_06.csv` which lists the Covid-19 cases per country per date similar to listing the population data.
- Modify `population_data.hpp` and `population_data.cc` to instead load the data from the Covid19 data file. You should have a structure named `Covid19Data` and it should have a field for each column. You can treat dates as strings. Save the resulting code in `covid19_data.hpp` and `covid19_data.cc`.
- You can use any one of the sort algorithms we have used in the exercises including Quicksort that you will implement in Exercise 01. Copy one of these into the `src` directory.
- Create a new C++ file `covid19_list.cc` and place the `main` function in that file. You can reuse code from the source file `sort_test.cc`.
- Finish the implementation such that it has the following functionality:
  - It takes the name of the data filename as its first argument. Loads the contents of the file as a vector of `Covid19Data` objects.
  - By default, it sorts the data items by new cases and displays each field of the topmost 10 data items, including the date and the location.
  - If given a command line option "`--by-total`" anywhere after the data filename, it sorts by total cases.

  - If given a command line option "`--by-deaths`" anywhere after the data filename, it sorts by new deaths.
  - If given a command line option "`--by-total`" and "`--by-deaths`" anywhere after the data filename, it sorts by total deaths.
  - If given a command line option "`-n`" followed by a positive integer N, it lists the N topmost items instead of just 10. If $N < 1$, you should display an error and quit.

- The usage message should look like

```
Usage: ./src/covid19-list <data-filename> [options]
Options:
  --by-total  : sort by totals instead of new
  --by-deaths : sort by deaths instead of cases
  -n <N>      : display top N items instead of 10
```

- The output from the command

  `./src/covid19-list ../data/covid_data_full_2020_04_06.csv --by-deaths -n 15`

  must look like

```
   Date   | New Cases|New Deaths| Total Cases|Total Deaths|Location
----------|----------|----------|------------|------------|--------
2020-04-04|      5233|      2004|       64338|        6507|France
2020-04-05|     34272|      1344|      312237|        8501|United States
2020-04-06|     25398|      1146|      337635|        9647|United States
2020-04-04|     32425|      1104|      277965|        7157|United States
2020-04-02|     27103|      1059|      216721|        5138|United States
2020-04-05|      4267|      1053|       68605|        7560|France
2020-03-28|      5959|       971|       86498|        9136|Italy
2020-04-03|      8102|       950|      110238|       10003|Spain
2020-04-04|      7472|       932|      117710|       10935|Spain
2020-04-03|     28819|       915|      245540|        6053|United States
2020-04-01|     24998|       909|      189618|        4079|United States
2020-03-29|      5974|       887|       92472|       10023|Italy
2020-04-02|      7719|       864|      102136|        9053|Spain
2020-04-01|      9222|       849|       94417|        8189|Spain
2020-04-01|      4053|       839|      105792|       12430|Italy
```

  including the alignment of data items.

Exercise 2 Improving Efficiency (30/100 pts)

To display the topmost N items, we need not sort the entire data set. Improve the efficiency of your algorithm by finding the topmost N items using a variant of the Quick-select algorithm. Quick-select only returns the N^{th} item, think about how to modify it to compute the topmost N items.

Exercise 3 The Bonus (+10 pts)

Implement the first part of the homework in standard Python. Place your implementation in the directory `ceng218_hw01/py`.