INSTRUCTIONS

1. How to Compile (on Windows, for example)

Install MinGW-W64 GCC on Windows, and enter in Command Prompt

```
g++ md.cpp lb.cpp dtw.cpp pnt.cpp
```

which will generate an executable named `a.exe`, or enter in Command Prompt

```
g++ md.cpp lb.cpp dtw.cpp pnt.cpp -o any_name_you_like.exe
```

which will generate an executable `any_name_you_like.exe`

Note: Our version of g++ provided by MinGW-w64 GCC is 7.1.0. If you have MinGW-W64 GCC installed, and enter in Command Prompt

```
g++ --version
```

the machine will print on screen something like

```
g++ (x86_64-posix-seh-rev2, Built by MinGW-W64 project) 7.1.0
```

2. How to Use (on Windows, for example)

Examples

(1) Assume the executable's name is `a.exe`. Enter in Command Prompt

```
a
```

then the program will run on a self-generated random time series with default parameters.

(2) Assume you have a data file `eeg400` (*included*) in the `data` folder. Enter in Command Prompt

```
a data\eeg400
```

then the program will run on `eeg400` with default parameters.

(3) Assume you have a data file `ox866` (*included*) in the `data` folder. Enter in Command Prompt

```
a data\ox866 2
```

then the program will run on `ox866` with Sakoe-Chiba band $r = 2$.

(4) Assume you have a data file `light1189` (*included*) in the `data` folder. This data has 1,189 points. You don't want that long. Enter in Command Prompt

```
a data\light1189 1000 1
```

then the program will run on the first 1,000 data points with Sakoe-Chiba band $r = 1$.

(5) Assume you have a data file `sea1400` (*included*) in the `data` folder. This data has 1,400 points. You want the subsequence length m to be 200. Enter in Command Prompt

```
a data\sea1400 1400 1 200
```

then the program will run with Sakoe-Chiba band r = 1, and subsequence length m = 200.

(6) Assume you have a data file `forex1000` (*included*) in the `data` folder. This data has 1,000 points. You want the subsequence length m to be 5% of total length. Enter in Command Prompt

```
a data\forex1000 1000 1 0.05
```

then the program will run with Sakoe-Chiba band r = 1, and subsequence length m = 0.05 * 1000 = 50.

Command Line Options

```
# 1: [command]

# 2: [command] [filename]

# 3: [command] [filename] [band width r]

# 4: [command] [filename] [length n] [band width r]

# 5: [command] [filename] [length n] [band width r] [subsequence length m]

# 6: [command] [filename] [length n] [band width r] [subsequence proportion mm]
```

Requirements

```
# Require Time Series Data Length n > 100

# Require Sakoe-Chiba Band r in [0, m/2]

# Require Subsequence Length m in [1, n/2], or Subsequence Proportion mm in (0, 0.5]

# Max Time Series Data Length n = 10000

# Default Time Series Data Length n = 1000

# Default Sakoe-Chiba Band r = 1

# Default Subsequence Length m = n/10, or Subsequence Proportion mm = 0.1

# If No Data File (#1), Program will Generate a Random Time Series.
```

3. Description of Source Files and Dataset

Source Files

`md.hpp`	header for `md.cpp`
`lb.hpp`	header for `lb.cpp`
`dtw.hpp`	header for `dtw.cpp`
`pnt.hpp`	header for `pnt.cpp`
`mat.hpp`	template functions for [**MAT**]rix
`md.cpp`	[**M**]atrix profile with [**D**]ynamic time warping, the main source file
`lb.cpp`	[**L**]ower [**B**]ound functions
`dtw.cpp`	[**D**]ynamic [**T**]ime [**W**]arping functions
`pnt.cpp`	[**P**]ri[**NT**]ing functions

Dataset

`eeg3600`	EEG (electroencephalogram) recordings. 3,600 data points.
`eeg400`	EEG (electroencephalogram) recordings. 400 data points.
`forex1000`	daily USD/GBP exchange rates. 1,000 data points.
`light1189`	10-day mean light intensity recordings from S Carinae star. 1,189 data points.
`ox866`	oxygen-18 to oxygen-16 ratio in about 2.5 million years. 866 data points.
`sea1400`	Darwin Sea level pressures (monthly), from 1882 to 1998. 1,400 data points.
`soi540`	the Southern Oscillation Index, related to climate change. 540 data points.

** This dataset is acquired from the Department of Statistical Science of Duke University.*

http://www2.stat.duke.edu/~mw/ts_data_sets.html

4. An Example to Show How to Run the Program

Assume the executable's name is `a.exe`, and you have a data file `light1189` (*included*) in the `data` folder. Enter in Command Prompt

```
a data\light1189 1000 1 0.05
```

The program will run on the first 1,000 data points of `light1189`, with Sakoe-Chiba band r = 1, and use 5% of the total length, which is 50, as the query subsequence length. Shortly it will print on screen

```
Length of Time Series Data n = 1000
Length of Subsequence m = 50
Length of Matrix Profile l = 951
Length of Forbidden Zone f = 13
Entries to Compute g = 880782
Sakoe-Chiba Band r = 1
DTW Brutal Force : Time = 6.2 s
Lower Bound Direct : Time = 2.0 s
Lower Bound Incremental : Time = 0.7 s
::::Lower Bound Consistency Check Begins:::::::
::::Lower Bound Consistency Check Ends:::::::::
::::Lower Boundedness Check Begins:::::::::::::
::::Lower Boundedness Check Ends:::::::::::::::
DTW Lower Bound : Time = 0.4 s
# Saved = 857231 (97%)
::::Matrix Profile Consistency Check Begins::::
::::Matrix Profile Consistency Check Ends::::::
DTW Randomized : Time  = 1.1 s
# Saved = 864752 (98%)
DTW Simulated Annealing : Time  = 1.2 s
# Saved = 864133 (98%)
::::Matrix Profile Consistency Check Begins::::
::::Matrix Profile Consistency Check Ends::::::
```

This tells you, after printing the basic information, line by line,

(1) the run time of computing DTW by brutal force for the entire matrix (6.2 s);

(2) the run time of directly computing the lower bounds for the entire matrix (2.0 s);

(3) the run time of incrementally computing the lower bounds for the entire matrix (0.7 s);

(4) the two sets of lower bounds are the same (pass the consistency check);

(5) the run time of computing DTW sequentially with the help of lower bounds (0.4 s);

(6) the number (857231) and percentage (97%) of entries saved from expensive computation of DTW;

(7) the two sets of matrix profiles are the same (pass the consistency check);

(8) the run time of computing DTW in a randomized way with the help of lower bounds (1.1 s);

(9) the number (864752) and percentage (98%) of entries saved from expensive computation of DTW;

(10) the run time of computing DTW with simulated annealing with the help of lower bounds (1.2 s);

(11) the number (864133) and percentage (98%) of entries saved from expensive computation of DTW;

(12) the two sets of matrix profiles are the same (pass the consistency check).

5. More Information

Naming of Variables in Source Code

```
t – time series

n – length of entire time series data

m – query length of subsequence

l – length of matrix profile, which is (n – m + 1)

f – length of forbidden zone, where there is no need to compute

g – number of entries to compute (all entries in matrix except for the forbidden)

r – Sakoe-Chiba band

e – random engine

...
```

** The `main()` functions commented out in `dtw.cpp` and `lb.cpp` alone can be used as test functions for the individual source files. Indeed, I (Fu Lei) implemented more functions than what are printed on screen. For example, setting `hdebug` to `true` may let the program to print out the optimal path of the DTW like* (0,0) → (0,1) → (0,2) →(1,2)→(2,3)→(3,3).