CS202 - Programming Assignment 1 - Sorting
Due: 23:55, June 26 (Wednesday), 2013.

June 20, 2013

1 Introduction

In this assignment, you should implement a simple version of UNIX `sort` command (reading material: http://bit.ly/odu1wE). Moreover, you will learn how to use basic UNIX commands (`time`, `diff`, etc).

2 Input

- Given a text file with N **lines** ($N \leq 100000$).
- Every line of the file contains same number of **words** (at most 100 words).
- Total number of lines and total number of words per line is variable, but it is guaranteed that the size of the file won't exceed 250 MB.
- Lines are separated using `new line` character ('\n').
- Words on the same line are separated using `space` character (' ').
- Any word in the file obeys one of the following rules,
  - contains only lowercase letters of the English alphabet ('a'-'z')., i.e., it's a ***dictionary word***
  - contains only digits ('0'-'9'), i.e., it's a ***number***
- If a word obeys the second rule (i.e., if it's a number), its numeric value will be **positive** and **less than** 2^{30}, i.e., it fits into built-in `int` type in C++. Moreover, it won't start with digit `'0'`.
- j^{th} *column* of the file is set of words that are located on j^{th} positions of lines. Formally,

$$C_j = \{w_{ij} \,|1 \leq i \leq N\}\,.$$

  where C_j stands for "j^{th} column of the file" and w_{ij} stands for "j^{th} word of i^{th} line of the file".

- To clarify, let's work on a sample file:

```
naturalistic damageable 10 prediscontinuance
coppering noninterpolation 20 nonvalidity
artless baya 30 clerestory
cyanhydrate monotrichous 40 escape
keel antennate 50 withouten
psychotechnical barricader 60 knightia
mantidae nonarrival 70 mate
spinsterishly reed 80 asparagine
mahseer pet 90 pyrenodeine
unornly superpolitic 100 corkscrewy
```

  1^{st} column of the sample file:

```
naturalistic
coppering
artless
cyanhydrate
keel
psychotechnical
mantidae
spinsterishly
mahseer
unornly
```

  2^{nd} column of the sample file:

```
damageable
noninterpolation
baya
monotrichous
antennate
barricader
nonarrival
reed
pet
superpolitic
```

  3^{rd} column of the sample file:

```
10
20
30
40
50
60
70
80
90
100
```

4^{th} column of the sample file:

```
prediscontinuance
nonvalidity
clerestory
escape
withouten
knightia
mate
asparagine
pyrenodeine
corkscrewy
```

- A column is composed of either numbers or dictionary words. That is, a column won't contain both numbers and dictionary words together.
- All numbers or dictionary words in the same column are unique.

3 Assignment

In this assignment, you job is to sort all lines in input text file in ascending/descending order of j^{th}column. If j^{th} column contains numbers, then you should implement a *numeric sorting*. If j^{th} column contains dictionary words, then you should implement a *lexical/lexicographical sorting* (search for *lexicographical order* on the web). Some clarifying examples are given below:

Sort 1^{st} column in *ascending* order (first column is sorted in ascending lexicographical order):

```
artless baya 30 clerestory
coppering noninterpolation 20 nonvalidity
cyanhydrate monotrichous 40 escape
keel antennate 50 withouten
mahseer pet 90 pyrenodeine
mantidae nonarrival 70 mate
naturalistic damageable 10 prediscontinuance
psychotechnical barricader 60 knightia
spinsterishly reed 80 asparagine
unornly superpolitic 100 corkscrewy
```

Sort 1^{st} column in *descending* order:

```
unornly superpolitic 100 corkscrewy
spinsterishly reed 80 asparagine
psychotechnical barricader 60 knightia
naturalistic damageable 10 prediscontinuance
mantidae nonarrival 70 mate
mahseer pet 90 pyrenodeine
keel antennate 50 withouten
cyanhydrate monotrichous 40 escape
coppering noninterpolation 20 nonvalidity
artless baya 30 clerestory
```

Sort 3^{rd} column in *descending* order (third column is sorted in descending numeric order):

```
unornly superpolitic 100 corkscrewy
mahseer pet 90 pyrenodeine
spinsterishly reed 80 asparagine
mantidae nonarrival 70 mate
psychotechnical barricader 60 knightia
keel antennate 50 withouten
cyanhydrate monotrichous 40 escape
artless baya 30 clerestory
coppering noninterpolation 20 nonvalidity
naturalistic damageable 10 prediscontinuance
```

4 Output

Output (sorted lines) should be printed into an output file in exactly same format as explained in Section 2.

5 Remarks

- Your program should read, sort and write the given text file in less than 1 minute. Any implementation which runs for more than 1 minute will **receive 0**. Therefore, sorting algorithms with O(N^2) run-time complexity won't help you this time. You should think about something faster, let's say with run-time complexity of O($NlogN$). You can also design and implement your own sorting algorithm and name it *AhmetMehmetSort*. Just, make sure that it reads, sorts and writes in less than 1 minute in the worst case.

- Your source code should be in a file named `hw1.cpp`. It's `main` function should take 4 command line arguments.

  - 1^{st} argument is the name of the input file.
  - 2^{nd} argument is the name of the output file.
  - 3^{rd} argument is a number which specifies the number of the column to be sorted.
  - 4^{th} argument is either `desc` or `asc`. If its `desc`, then you should sort in descending order. Otherwise, you should sort in ascending order.

  Sample execution:
  ```
  ./hw1 input.txt output.txt 3 desc
  ```

- You source code will be compiled on **dijkstra**(if you have any problems using **dijkstra** machine, send an e-mail to shatlyk@cs.bilkent.edu.tr ASAP) with the following command:
  ```
  g++ -Wall hw1.cpp -o hw1
  ```
  Any solution that does not compile will **receive 0**. No excuses.

- After compilation, the compiled `hw1` will be executed with the following command:
  ```
  time ./hw1 input_filename output_filename column_number order
  ```
  Output should be written into the file with name `output_filename`. No output should be written into `stdout`.

- `time` command outputs the total run-time of the program. Read the documentation of `time` command for details.

- The correctness of the produced output file `output_filename` will be checked using `diff` command:
```
diff output_filename correct_output_filename
```
This command should **not** produce any output, which means that 2 files are identical. Read the documentation of `diff` command for details.

- For this part of the assignment you are **not** allowed to use C++ STL or any other library that presents sorting functions (qsort() in C, std::sort() in algorithm.h, etc). Moreover, you are **not** allowed to used any containers from C++ STL.

- Your code should make dynamic memory allocation. Which means, **static arrays with predefined sizes are not allowed.**

- Be careful with memory leaks. We will profile your solutions for memory leaks.

- Compress **only** hw1.cpp into **hw1.tar.gaz** archive using `tar` command in UNIX. Any archive such as hw1.zip, hw1.rar, hw1.*xxx*, hw1_Ahmet_Mehmet.tar.gz will **receive 0**. No excuses! Do not compress using Winzip or 7zip or any other Windows software! Do not compress into .zip archive and rename the extension to .tar.gz!

- Since we grade your solutions using grader program, any simple error will end up with 0 points.

- If you are undecided about any part, do not make your own assumptions. Instead ask to TA.

6 Cheating

Your professor and TA are willing to help you 7/24. You may ask for any kind of help by sending an e-mail to shatlyk@cs.bilkent.edu.tr. We appreciate such students and do our best to push them forward. So, do all of your assignments by yourselves. Data Structures course is one of the main core courses of CS. So, try to build a strong CS background by doing assignments by yourselves.

If the last paragraph does not convince you not to cheat, here are some details about cheating detection: Your source code will be checked against cheating very strictly. We use cheating detection software that detects more than 99% of the cheaters. Changing the function name, variable names, etc won't help you this time. Believe us. If you feel lucky enough to be in that 1% portion, just go ahead. Our last advice: don't lead yourself into trouble.

7 Contact

You can ask your questions to Shatlyk Ashyralyyev (shatlyk@cs.bilkent.edu.tr).