Hersh Patel
10/6/2017

Project Documentation

What's Included:

`ec2.c`

- Main Method
    - Parameter Error Checking
    - Read header into header_data struct array
    - Read data into row_data struct array
    - Call mergesort on data with respective column name
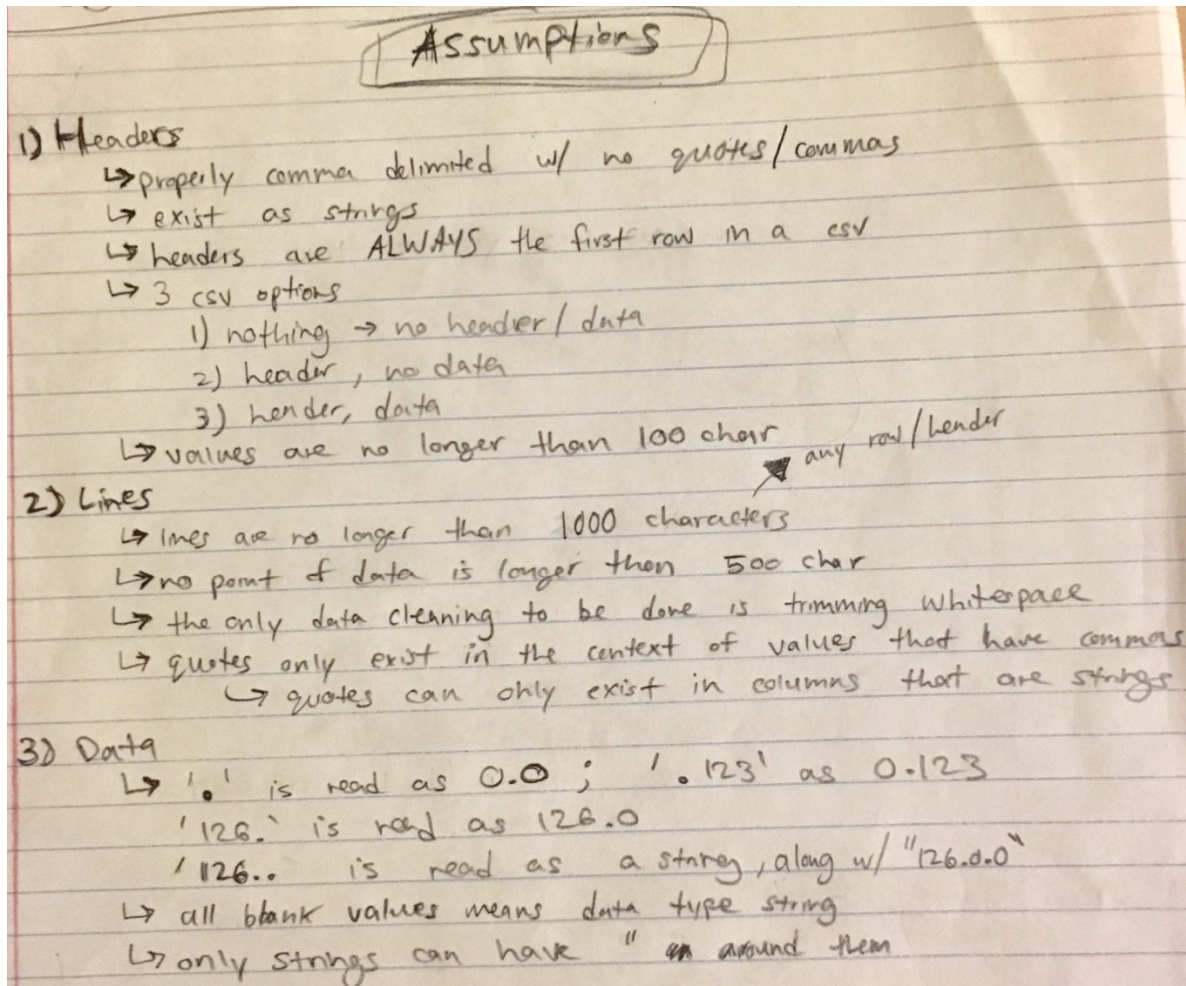   - Print out the data correctly

`ec2.h`

- `int num_each_type` →
    - array of the number of each type of data in each row
    - index 0 (# of strings), index 1 (# of ints), index 2 (# of doubles), index 3 (# of blanks)
- `int getDataType(char* token)` →
    - returns -1 for blanks, 0 for strings, 1 for ints, 2 for doubles
- `int hasQuote(char* token)` →
    - returns 1 if given string has a quote and 0 if it does not
- `struct header_data` →
    - holds all relevant data for the header of a dataset
- `struct row_data` →
    - holds all relevant data for the row of a dataset

`ec2_mergesort.c`

- `void merge` →
    - sorts strings
- `void merge1` →
    - sorts ints
- `void merge2` →
    - sorts doubles

Assumptions:

- *There only exists 3 types of data* (`strings, ints, and doubles`): Our algorithm only distinguishes between these 3 data types, assigning columns to these data types

Assumptions

1) Headers
   - ↳ properly comma delimited w/ no quotes/commas
   - ↳ exist as strings
   - ↳ headers are ALWAYS the first row in a csv
   - ↳ 3 csv options
     1) nothing → no header/data
     2) header, no data
     3) header, data
   - ↳ values are no longer than 100 char ↗ any row/header

2) Lines
   - ↳ lines are no longer than 1000 characters
   - ↳ no point of data is longer than 500 char
   - ↳ the only data cleaning to be done is trimming whitespace
   - ↳ quotes only exist in the context of values that have commas
     - ↳ quotes can only exist in columns that are strings

3) Data
   - ↳ '.' is read as 0.0 ; '.123' as 0.123
   - '126.' is read as 126.0
   - '126..' is read as a string, along w/ "126.0.0"
   - ↳ all blank values means data type string
   - ↳ only strings can have " around them

Algorithm:

1) Dynamically read in the header row
   a. Determine the # of columns that exist in dataset
2) Read in all of the data as strings
   a. Determine the # of rows that exist in a dataset
3) Distinguish column data type
4) Re-read data and convert each field to its proper data types
5) Perform mergesort on the selected column
6) Print out the sorted csv, maintaining original order of columns