Group Programming Project User’s Manual

Dual Heap Replacement Selection Sort

By: Jared Rickert, George Velikson, Kelden Mayer, Chris Vadner, and Sanju Neupane

This program was designed to perform a replacement selection sorting technique using a dual heap abstract data type. The program reads in a file of unsorted records and uses sorting techniques and algorithms to sort output the records into a new file that consists of the records in a sorted form based on which keys the user entered.

When the program is finished, there is also a log file that can be generated that includes statistics retrieved from the program’s runs.

The files to the program are in the folder named /dual-heap/ and are located in the following path:

/export/home/csci331/cs331xx/dual-heap/

To compile and the program, the user must simply type:

> make build

Once the program is running, the user will see a blank line that will accept the user instructions for the program.

The interface of the program will have multiple flags that are either required or optional for use.

**-v** *(optional)* This flag is used for debugging purposes. The syntax is:

>/bin/program <FILE\_BEING\_READ\_FROM\_PATH.csv> **-o <OUTPUT\_FILE.csv> -v**[ENTER]

**-o** *(required)* This flag is used to specify the name of the file the program is outputting to. The syntax is:

>/bin/program <FILE\_BEING\_READ\_FROM\_PATH.csv> **-o <OUTPUT\_FILE.csv>**[ENTER]

**-l** (*optional)* This flag is used when the user would like the program to output a log file. The syntax is:

>/bin/program <FILE\_BEING\_READ\_FROM\_PATH.csv> -o <OUTPUT\_FILE.csv> **-l <LOG\_FILE.txt>**[ENTER]

**-k** *(optional)* This flag is used to sort by a specific key in the .csv file. The syntax is as follows:

>/bin/program <FILE\_BEING\_READ\_FROM\_PATH.csv> -o <OUTPUT\_FILE.csv> **-k <KEY1,KEY2(optional)>**[ENTER]

By default, if nothing is entered, this program will read from the single first key the program sees and output the results from that.

Once the program is finished, it will exit and output any files specified by the user.

Known errors:

- Unformatted input will result in undefined behavior.

- No output file will result in the program exiting upon initiation.

- If there is no temp directory made then the program can’t find it, the program will silently fail and there will be unusual errors.