CPE 212 Laboratory 9

We have discussed a variety of sorting algorithms and analyzed their behavior. For situations where you have a moderate amount of data these algorithms are fine.

However, sorting usually involves huge amounts of data, far more than can be stored in the memory of your computer. For example several hundred million tax returns each year.

The algorithms you have are particularly sensitive to this problem as much of their data storage is allocated on the stack which is quite limited.

The approach that is usually taken in these cases is to build a merge sort that performs the merging on disk files which can be almost arbitrarily long.

- For this lab,
- 1- create a data file of 50 million 3 digit random numbers.
- 2- Test two existing sorting algorithms to determine how many data items they can sort. Measure and report their time using the functions described in the file fine_grain_time.cpp
- 3- divide the 50 million items into as many files as is required to perform the sort on each of the individual files. For example if the sort algorithm you chose can sort 1 million items, then create 50 separate files and sort each file individually. Under no conditions should you have less than 10 files even if you can.
- 4- merge each of individually sorted files into one final sorted file with 50 million items.
- 5- To verify that your work is correct, compare your final result to that obtained by applying the sort command (man 1 sort) to your original file and see if they agree (man 1 diff).

- For the base sorting methods (that sort the data in memory) measure the time to sort using the code in the file “fine_grain_time.cpp” in the lab9 files directory

- Split command to split 50 million file
- Sort each file separately using whatever sort file you want
- Merge the files back together again.
- His algorithm take 58 seconds to complete
- Open all 10 files then read in the first number