William Horn

Description:

This project required us to read in from a file specified by the user to populate a vector of Employee classes. For debugging purposes, I made “debug.dat” be one of the possible good inputs. Debug.dat was a small sample of data that would be faster to run than the full records.dat file. The Employee class holds three variables. The employees ID number, their last name and their salary. We then had to take this vector of Employees and use it to populate two ordered and two unordered maps. The keys for the two types of maps were the same. One of the keys for each was the employees department number, which is also the first four digits of their ID number, and the employees salary in tens of thousands. We had to time the speed that each of the maps were created in terms of clock ticks, and we also had to output the largest salary range and how many employees were in said range.

Manifest:

- Project2.pdf
- empdriver.cpp
- Employee.cpp (I put my name at the top, but otherwise it is not changed)
- Employee.h (I put my name at the top, but otherwise it is not changed)
- empmaps.cpp
- empmaps.h

Questions:

a. The runtime in clock ticks for creating the ordered map with departments as the key was 144. The runtime in clock ticks for creating the unordered map with departments as the key was 87. I think that the unordered_map was faster at this because all it had to do was populate the unordered_map, whereas the ordered map had to populate the map and then organise it into the proper order.
b. The runtime in clock ticks for creating the ordered map with salary as the key was 69. The runtime in clock ticks for creating an unordered map with salary as the key was 44. I think that the unordered_map was faster for the same reason as with the key being the department number. The unordered_map only has to be populated, whereas the ordered map had to be populated and then sorted, therefore adding to the runtime.
c. When I run the code using records2.dat rather than records.dat, the runtimes all went up. The runtime in clock ticks for creating the ordered map with departments as the key was 267 and the runtime in clock ticks for creating the unordered_map with departments as the key was 145. The runtime in clock ticks for creating the ordered map with salary as the key was 150 and the runtime in clock ticks for creating the unordered_map with salary as the key was 94. Obviously the clock times went up as we had double the data, and they seemed to increase at a linear rate, although they were a bit less than linear, this would indicate to me that the runtime for creating both of these maps is O(n). That is the main difference in the data. Other than that, the ordered map and the unordered_map both have the same number of employees in each of the salary ranges,

they also have the same number of salary ranges and departments as they did when they were made using records.dat because records2.dat contained the same entries, just double the number of them.