

# Homework: Sorting on External Memory (w7)

[New Attempt](#)

**Due** May 24 by 11:59pm **Points** 8 **Submitting** a file upload

## Introduction

The objective of this assignment is to learn the implementation of sorting algorithms on external memories.



## What you must do

Consider the following relations:

```
Emp (eid (integer), ename (string), age (integer), salary (double))
```

Fields of types integer, double, and string occupy 4, 8, and 40 bytes, respectively. Each page (block) can fit at most one tuple (record) of the input relation. Assume that each page (block) can fit at most one record (tuple) of the input file. Implement the two-pass multi-way sorting for the file **Emp.csv** in C/C++ using the skeleton code posted with this assignment. The sorting should be based on the attribute **eid**. There are at most 22 pages available to the sort algorithm in the main memory, i.e., the size of the buffer is 22

- The input relation is stored in a CSV file, i.e., each tuple is in a separate line and fields of each record are separated by commas. The file that store relation Emp is Emp.csv.
- The result of the sort must be stored in a new CSV file. The file that stores the relation Emp is Emp.csv.
- Your program must assume that the input file is in the current working directory, i.e., the one from which your program is running.
- The program must store the result in a new CSV file with the name EmpSorted.csv in the current working directory.

- Your program must run on Linux. Each student has an account on `hadoop-master.engr.oregonstate.edu` server, which is a Linux machine. You may use this machine to test your program if you do not have access to any other Linux machine. You can use the following bash command to connect to it:

```
> ssh your_onid_username@hadoop-master.engr.oregonstate.edu
```

Then it asks for your ONID password and probably one another question. You can access this server on campus. If you want to access from outside of campus, you'll need to use VPN to access the campus network.


- You must name the file that contains the source code of the `main()` function `main4.cpp`.
- You may use following commands to compile and run C++ code:

```
> g++ -std=c++11 main4.cpp -o main4.out > main4.out
```


There is no need to use double buffering in your implementation.


## Necessary files

---

Input file: **Emp.csv** (<https://canvas.oregonstate.edu/courses/1939345/files/99423795?wrap=1>)   
([https://canvas.oregonstate.edu/courses/1939345/files/99423795/download?download\\_frd=1](https://canvas.oregonstate.edu/courses/1939345/files/99423795/download?download_frd=1))

Following files contain the skeleton code to generate the required `EmpSorted` file. You may make changes to these files adhering to the assignment requirements.

**main.cpp** (<https://canvas.oregonstate.edu/courses/1939345/files/99252773?wrap=1>)   
([https://canvas.oregonstate.edu/courses/1939345/files/99252773/download?download\\_frd=1](https://canvas.oregonstate.edu/courses/1939345/files/99252773/download?download_frd=1))

**record\_class.h** (<https://canvas.oregonstate.edu/courses/1939345/files/99252774?wrap=1>)   
([https://canvas.oregonstate.edu/courses/1939345/files/99252774/download?download\\_frd=1](https://canvas.oregonstate.edu/courses/1939345/files/99252774/download?download_frd=1))

## What to turn in

---

The assignment is to be turned in before Midnight (by 11:59pm). You may turn in the source code of your program through Canvas. The assignment may be done in groups of two students. Each group may submit only one file that contains the full name, OSU email, and ONID of every member of the group.

## Grading criteria

---

The programs that implement the correct algorithm, return correct answers, and do not use more than allowed buffers will get the perfect score. The ones that use more buffer than allowed will not get any points. The ones that implement the right algorithm but return partially correct answers will get partial credits.

Sorting on External Memory					
Criteria	Ratings				Pts
Description of criterion	<b>8 pts Full Marks</b>	<b>6 pts Minor Error</b> Sorted result missing some tuples.	<b>4 pts Major Error</b> You did not sort into Runs and then use those Runs to generate the final sorted file.	<b>0 pts No Marks</b> You did not meet the assignment requirements	8 pts
					Total Points: 8