CS315: Assignment on External Mergesort

Marks = 50

Deadline: 21st March, 2023 at 7:30am

Implement/simulate the **external mergesort** algorithm.

Input

Your program should take the following inputs:

- file containing the keys (each line contains a single key),
- size of available memory, m, in number of blocks, = ram size= gfg run size
- size of each key, k, in bytes,
- total number of keys, n, and
- disk block size, b, in bytes.

Assume an infinite disk size.
Ensure that your program reads these 4 integer inputs in order after the file that contains the keys.
An example of running the program is

```
./program-name input-file.txt 10 8 10000 1024
```

Disk Access

You may either implement or simulate the program.
You may implement disk access by actually writing to files in the O/S. Each file has the size of a disk block.
You may otherwise simulate the entire program in memory by counting the number of such disk reads/writes.
Keep a count of the total number of disk seeks and disk transfers.
Please remember that a random disk read/write incurs 1 disk seek and 1 disk transfer. A subsequent sequential disk read/write incurs only 1 disk transfer. The first block of a sequential read/write incurs 1 disk seek and 1 disk transfer.

Output

Your program should output the following:

1. The total number of disk seeks and disk transfers
2. The number of merge passes
3. The detailed output after the initial sorted run phase and each subsequent merge pass
4. The total cost (in terms of disk seeks and disk transfers) for the initial sorted run phase and each subsequent merge pass phase (and its sub-phases)

Submission

You should submit the entire running code (all the program files, Makefile, etc.) as a *single zip file*. Name your zip file as `rollno-mergesort.zip`.
After unzipping and compiling, it should produce an executable file that should run automatically with the input format as specified earlier.