

# **DSI Assignment 2.1 Report**

Shlok Gilda, UFID: 39548312, [shlokgilda@ufl.edu](mailto:shlokgilda@ufl.edu)  
Aditi Telang, UFID: 13277280, [atelang1@ufl.edu](mailto:atelang1@ufl.edu)

## **Compilation directions:**

We have implemented the necessary code. Use the following instructions to compile the code.

### **Run test (test.cc)**

1. make test.out
2. ./test.out

### **Run gtest (gtest.cc)**

1. make gtest.out
2. ./gtest.out

## **Function description:**

We have completed the BigQ class by writing the required functions. The job of the BigQFile within the system is to sort the records based on the input predicate, write to console and the store the sorted file with.bigq extension.

## **Implementation Details:**

We have created the following private fields in the BigQ.h.

```
Pipe *inputPipe; // Input pipe to get records.  
Pipe *outputPipe; // Output pipe to push records.  
OrderMaker *sortedOrder; // Sorted order required for sorting.  
int *runLength; // Run length.  
File *runsFile; // File pointer for the runs file.  
vector<int> runPointers; // List of pointers to all the runs.
```

## **Method Description:**

- **BigQ::BigQ(Pipe &in, Pipe &out, OrderMaker &sortorder, int runlen)**
  - This is the constructor of the BigQ class. A new “Runs” file is created to store the runs.

- **static void\* invoke\_tpmmsAlgo(void\* args)**
  - This function is the start of the worker thread. It performs the sorting. We are typecasting the void arguments to bigQ and invoking the worker function.
- **void BigQ::worker()**
  - Worker function which does the TPMMS Algorithm. The algorithm is written in two phases.
  - The first phase breaks the File into runs and sorts each run.
  - In the second phase, all the runs are merged into a single sorted file using priority queue.
- **void BigQ::sortRun(vector <Record\*> &)**
  - This function sorts the vector of records using the input predicate.
- **int BigQ::addRunToFile(vector <Record\*> &)**
  - This function takes input of sorted records into a vector and writes them to File and returns the current size of the file which acts as a pointer to the next run.

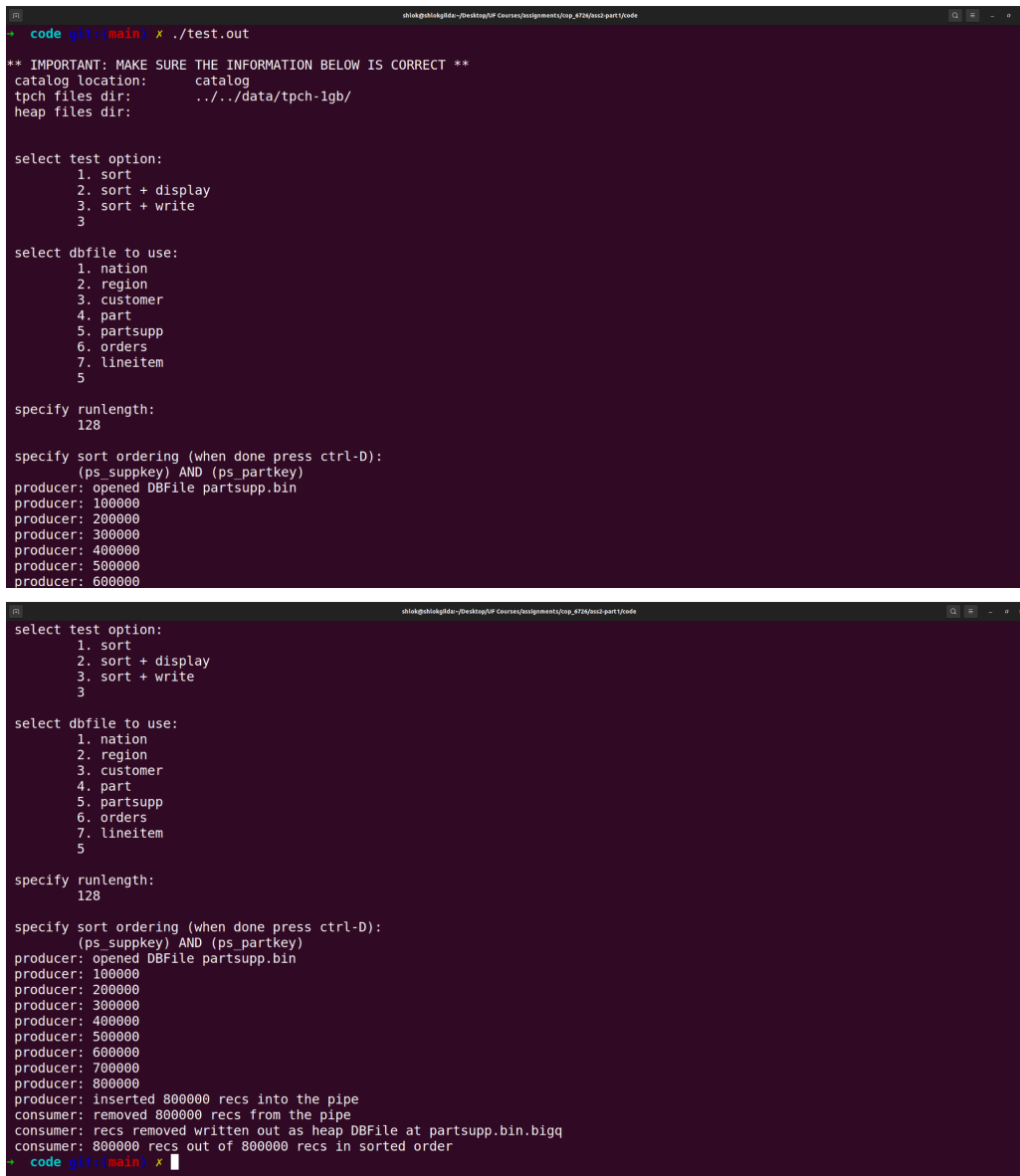
## **GTest test cases:**

### **Tests:**

1. Check whether the nation DB file is sorted or not.
2. Check whether the customer DB file is sorted or not.
3. Check whether the part DB file is sorted or not.
4. Check whether the lineitem DB file is sorted or not.

## Test case results:

1. The following is a screenshot of the output of “test.out”. The test case that we ran was:
  - a. **Test option:** 3. Sort + Write
  - b. **Dbfile:** 5. partsupp
  - c. **Runlength:** 128
  - d. **CNF:** (ps\_supkey) AND (ps\_partkey)



```
code git:(main) X ./test.out
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:      catalog
tpch files dir:       ../../data/tpch-1gb/
heap files dir:

select test option:
  1. sort
  2. sort + display
  3. sort + write
  3

select dbfile to use:
  1. nation
  2. region
  3. customer
  4. part
  5. partsupp
  6. orders
  7. lineitem
  5

specify runlength:
128

specify sort ordering (when done press ctrl-D):
(ps_supkey) AND (ps_partkey)
producer: opened DBFile partsupp.bin
producer: 100000
producer: 200000
producer: 300000
producer: 400000
producer: 500000
producer: 600000

select test option:
  1. sort
  2. sort + display
  3. sort + write
  3

select dbfile to use:
  1. nation
  2. region
  3. customer
  4. part
  5. partsupp
  6. orders
  7. lineitem
  5

specify runlength:
128

specify sort ordering (when done press ctrl-D):
(ps_supkey) AND (ps_partkey)
producer: opened DBFile partsupp.bin
producer: 100000
producer: 200000
producer: 300000
producer: 400000
producer: 500000
producer: 600000
producer: 700000
producer: 800000
producer: Inserted 800000 recs into the pipe
consumer: removed 800000 recs from the pipe
consumer: recs removed written out as heap DBFile at partsupp.bin.bigq
consumer: 800000 recs out of 800000 recs in sorted order
code git:(main) X
```

2. This is the output of running “./runTestCases.sh”. The “output1.txt” has been attached with the code. The screenshot below shows only a sample of “output1.txt”.

```
o_orderkey: [2199712], o_custkey: [66790], o_orderstatus: [0], o_totalprice: [515532], o_orderdate: [1996-09-30], o_orderpriority:
[2-HIGH], o_clerk: [Clerk#000000650], o_shippriority: [0], o_comment: [ the final, ironic deposits inte]
o_orderkey: [3586919], o_custkey: [24049], o_orderstatus: [F], o_totalprice: [522644], o_orderdate: [1992-11-07], o_orderpriority:
[1-URGENT], o_clerk: [Clerk#000000924], o_shippriority: [0], o_comment: [are alongside of the pending deposits. quick]
o_orderkey: [2232932], o_custkey: [13940], o_orderstatus: [0], o_totalprice: [522721], o_orderdate: [1997-04-13], o_orderpriority:
[2-HIGH], o_clerk: [Clerk#000000245], o_shippriority: [0], o_comment: [lithely stealthy accounts are slyly against the pend]
o_orderkey: [4576548], o_custkey: [108931], o_orderstatus: [0], o_totalprice: [525591], o_orderdate: [1997-12-26], o_orderpriority
: [1-URGENT], o_clerk: [Clerk#000000336], o_shippriority: [0], o_comment: [he slyly ironic requests. regular, bold deposits cajol]
o_orderkey: [3043270], o_custkey: [144617], o_orderstatus: [0], o_totalprice: [530604], o_orderdate: [1997-02-12], o_orderpriority
: [5-LOW], o_clerk: [Clerk#000000699], o_shippriority: [0], o_comment: [riously final deposits? ]
o_orderkey: [4722021], o_custkey: [128120], o_orderstatus: [F], o_totalprice: [544089], o_orderdate: [1994-04-07], o_orderpriority
: [1-URGENT], o_clerk: [Clerk#000000230], o_shippriority: [0], o_comment: [al, express pinto beans are after the careful]
o_orderkey: [1750466], o_custkey: [21433], o_orderstatus: [F], o_totalprice: [555285], o_orderdate: [1992-11-30], o_orderpriority:
[4-NOT SPECIFIED], o_clerk: [Clerk#000000040], o_shippriority: [0], o_comment: [ ironic packages. even notornis integrate. fluffi
ly regular foxes snooze furi]
consumer: removed 1500000 recs from the pipe
*****
n_nationkey: [7], n_name: [GERMANY], n_regionkey: [3], n_comment: [l platelets. regular accounts x-ray: unusual, regular acco]
n_nationkey: [19], n_name: [ROMANIA], n_regionkey: [3], n_comment: [ular asymptotes are about the furious multipliers. express dep
endencies nag above the ironically ironic account]
n_nationkey: [22], n_name: [RUSSIA], n_regionkey: [3], n_comment: [ requests against the platelets use never according to the quic
kly regular pint]
n_nationkey: [23], n_name: [UNITED KINGDOM], n_regionkey: [3], n_comment: [eans boost carefully special requests. accounts are. ca
refull]
n_nationkey: [4], n_name: [EGYPT], n_regionkey: [4], n_comment: [y above the carefully unusual theodolites. final dugouts are quic
kly across the furiously regular d]
n_nationkey: [10], n_name: [IRAN], n_regionkey: [4], n_comment: [efully alongside of the slyly final dependencies. ]
n_nationkey: [11], n_name: [IRAQ], n_regionkey: [4], n_comment: [nic deposits boost atop the quickly final requests? quickly regul
a]
n_nationkey: [13], n_name: [JORDAN], n_regionkey: [4], n_comment: [ic deposits are blithely about the carefully regular pa]
n_nationkey: [20], n_name: [SAUDI ARABIA], n_regionkey: [4], n_comment: [ts. silent requests haggle. closely express packages slee
p across the blithely]
consumer: removed 25 recs from the pipe
+ code git:(main) x
```

### 3. The following screenshots show the results of gtest.

```
shlok@shlokglide:~/Desktop/UF Course/assignments/top_8726/iss2-part1/code
+ code git:(main) x ./gtest.out
Gtest for sorted DBFile
[=====] Running 4 tests from 4 test suites.
[-----] Global test environment set-up.
[-----] 1 test from GTEST_1
[ RUN ] GTEST_1.SortedFileTestForNationTable
Gtest for checking whether the Nation DBFile is sorted or not
Reading nation.tbl records to input pipe....
Input pipe is filled with nation table records....
Sorting nation table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_1.SortedFileTestForNationTable (63 ms)
[-----] 1 test from GTEST_1 (63 ms total)

[-----] 1 test from GTEST_2
[ RUN ] GTEST_2.SortedFileTestForCustomerTable
Gtest for checking whether the Customer DBFile is sorted or not
Reading customer.tbl records to input pipe....
Input pipe is filled with customer table records....
Sorting customer table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_2.SortedFileTestForCustomerTable (471 ms)
[-----] 1 test from GTEST_2 (471 ms total)

[-----] 1 test from GTEST_3
[ RUN ] GTEST_3.SortedFileTestForPartTable
Gtest for checking whether the Part DBFile is sorted or not
Reading part.tbl records to input pipe....
Input pipe is filled with part table records....
Sorting part table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_3.SortedFileTestForPartTable (516 ms)
[-----] 1 test from GTEST_3 (516 ms total)

[-----] 1 test from GTEST_4
[ RUN ] GTEST_4.SortedFileTestForLineItemTable

[-----] 1 test from GTEST_2
[ RUN ] GTEST_2.SortedFileTestForCustomerTable
Gtest for checking whether the Customer DBFile is sorted or not
Reading customer.tbl records to input pipe....
Input pipe is filled with customer table records....
Sorting customer table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_2.SortedFileTestForCustomerTable (471 ms)
[-----] 1 test from GTEST_2 (471 ms total)

[-----] 1 test from GTEST_3
[ RUN ] GTEST_3.SortedFileTestForPartTable
Gtest for checking whether the Part DBFile is sorted or not
Reading part.tbl records to input pipe....
Input pipe is filled with part table records....
Sorting part table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_3.SortedFileTestForPartTable (516 ms)
[-----] 1 test from GTEST_3 (516 ms total)

[-----] 1 test from GTEST_4
[ RUN ] GTEST_4.SortedFileTestForLineItemTable
Gtest for checking whether the Lineitem DBFile is sorted or not
Reading lineitem.tbl records to input pipe....
Input pipe is filled with lineitem table records....
Sorting lineitem table records....
Completed Sorting input pipe records and pushed them to output pipe....
[ OK ] GTEST_4.SortedFileTestForLineItemTable (15688 ms)
[-----] 1 test from GTEST_4 (15688 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 4 test suites ran. (16738 ms total)
[ PASSED ] 4 tests.
+ code git:(main) x
```