HOMEWORK ASSIGNMNENT- 1

LARGE SCALE DATA STRUCTURES & ORG – INF 503

Name : Preetham John

NAU ID: 006060133

NAU Email: pj323@nau.edu

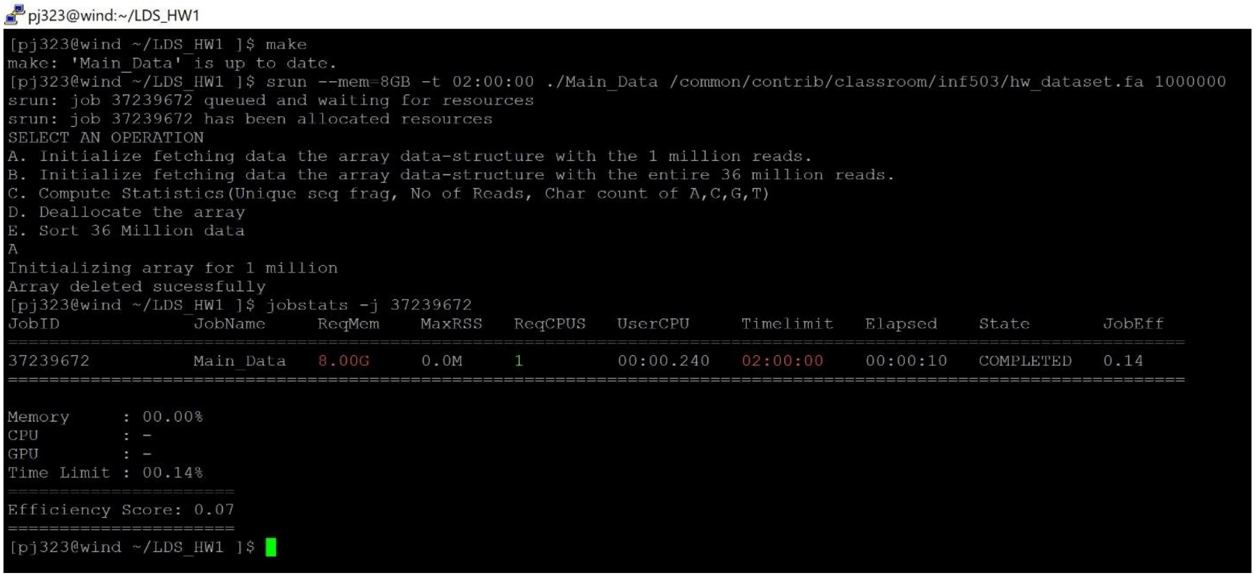
Northern Arizona University

Question A

Using the first 1 million reads, estimate, and report the total CPU time and RAM it will take to initialize (fill up) the array data-structure with the entire 36 million reads. Note that this may mean creating custom constructor to read first X reads rather than to the End-Of-File.

Initialized Job **ID = 37239672**

We re fetching 1 Million Reads or data and computing the system usage as asked in the Question above. Attaching a Screen shot of the same below.



```
[pj323@wind ~/LDS_HW1 ]$ make
make: 'Main_Data' is up to date.
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 02:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 1000000
srun: job 37239672 queued and waiting for resources
srun: job 37239672 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
A
Initializing array for 1 million
Array deleted sucessfully
[pj323@wind ~/LDS_HW1 ]$ jobstats -j 37239672
JobID             JobName     ReqMem    MaxRSS   ReqCPUS   UserCPU     Timelimit   Elapsed    State       JobEff
==================================================================================================================
37239672          Main_Data   8.00G     0.0M     1         00:00.240   02:00:00    00:00:10   COMPLETED   0.14
==================================================================================================================

Memory     : 00.00%
CPU        : -
GPU        : -
Time Limit : 00.14%
======================
Efficiency Score: 0.07
======================
[pj323@wind ~/LDS_HW1 ]$ 
```

Here we can see that the JOB was completed in **10 Seconds.** This shows nothing but 10 Seconds of the CPU usage time.

We can see that allocated RAM Size was **8GB,** but the used RAM was 0.0.

Manual Calculation

If 1000000 (1M) Data = 10 Seconds

Then for 36000000 (36M) Data = ?

On Manual calculation we know the general formula RAM = O(n)

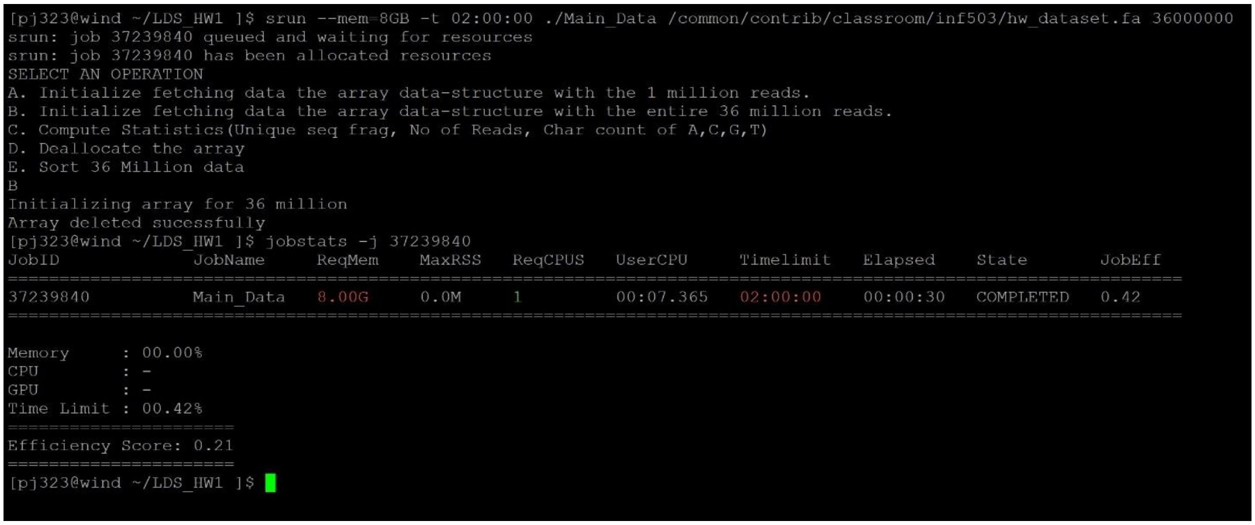
So we can use general formula and we find that it takes 3 Minutes for initializing 36 Million Data.

we can find the actual time it takes to do the same by initialising to fetch 36 Million data by running the Problem statement B.

Question B

Test your prediction using the entire 36 million read set – report actual RAM and CPU time used. Refer to Monsoon workshop notes for help in estimating actual runtime and RAM usage of your run. Were you accurate? If not, explain what you think caused the discrepancy.

Here we initialize to find the actual CPU time for initializing 36 Million Data and compare the statistics with the manual calculation.



```
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 02:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 36000000
srun: job 37239840 queued and waiting for resources
srun: job 37239840 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
B
Initializing array for 36 million
Array deleted sucessfully
[pj323@wind ~/LDS_HW1 ]$ jobstats -j 37239840
JobID             JobName     ReqMem    MaxRSS   ReqCPUS   UserCPU     Timelimit   Elapsed    State       JobEff
==================================================================================================================
37239840          Main_Data   8.00G     0.0M     1         00:07.365   02:00:00    00:00:30   COMPLETED   0.42
==================================================================================================================

Memory     : 00.00%
CPU        : -
GPU        : -
Time Limit : 00.42%
======================
Efficiency Score: 0.21
======================
[pj323@wind ~/LDS_HW1 ]$
```

To initialize 30M data it just took 30 Seconds

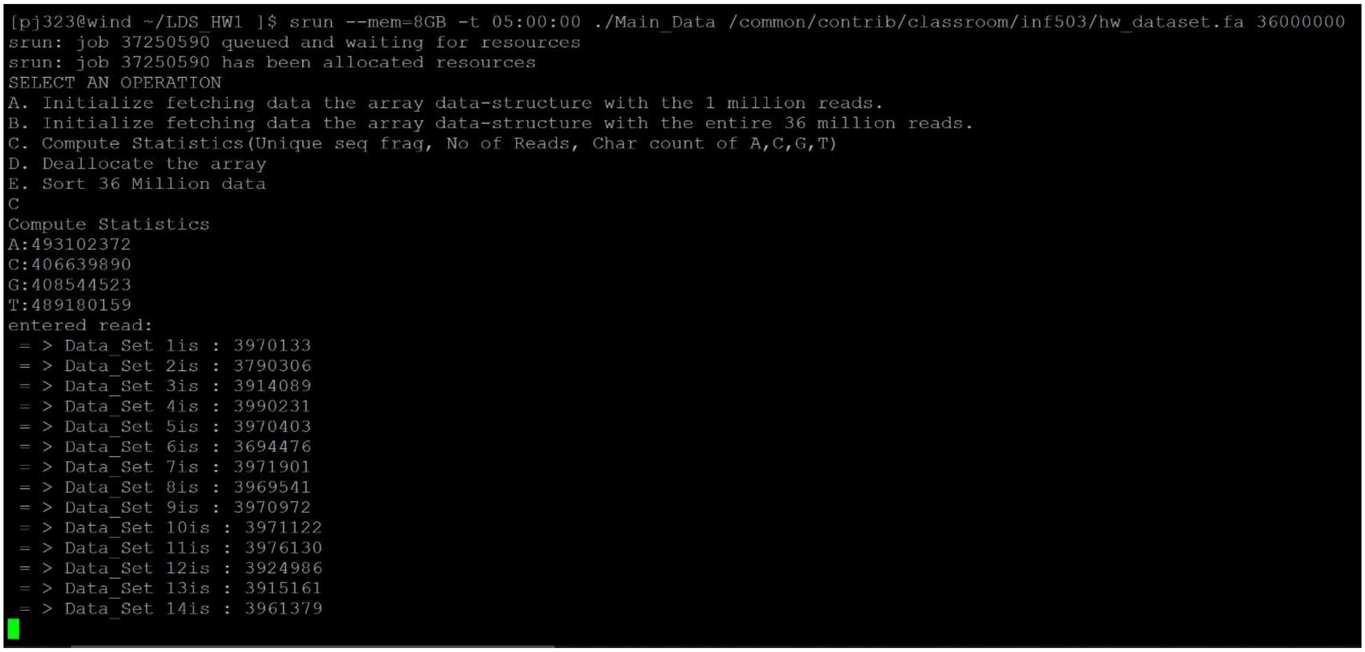
The Average CPU time used here is just = **30 Seconds** on a comparison it is faster and processing the more amount of data it just took 30 seconds.

Question C

Compute the following statistics for your read set

- Total number of unique sequence fragments (here, safe to assume this is the total number of sequence fragments in the file).
- Total number of reads for each ‘data set’ separately (recall there are 14 data sets in our example here). You will need 14 different totals.
- Number of A, C, G, and T characters in the dataset.

This is the Output when we tried to compute the data of **36000000 (36M) reads**. Was taking longer time for computation.



```
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 05:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 36000000
srun: job 37250590 queued and waiting for resources
srun: job 37250590 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
C
Compute Statistics
A:493102372
C:406639890
G:408544523
T:489180159
entered read:
 = > Data_Set 1is : 3970133
 = > Data_Set 2is : 3790306
 = > Data_Set 3is : 3914089
 = > Data_Set 4is : 3990231
 = > Data_Set 5is : 3970403
 = > Data_Set 6is : 3694476
 = > Data_Set 7is : 3971901
 = > Data_Set 8is : 3969541
 = > Data_Set 9is : 3970972
 = > Data_Set 10is : 3971122
 = > Data_Set 11is : 3976130
 = > Data_Set 12is : 3924986
 = > Data_Set 13is : 3915161
 = > Data_Set 14is : 3961379
```

The Count

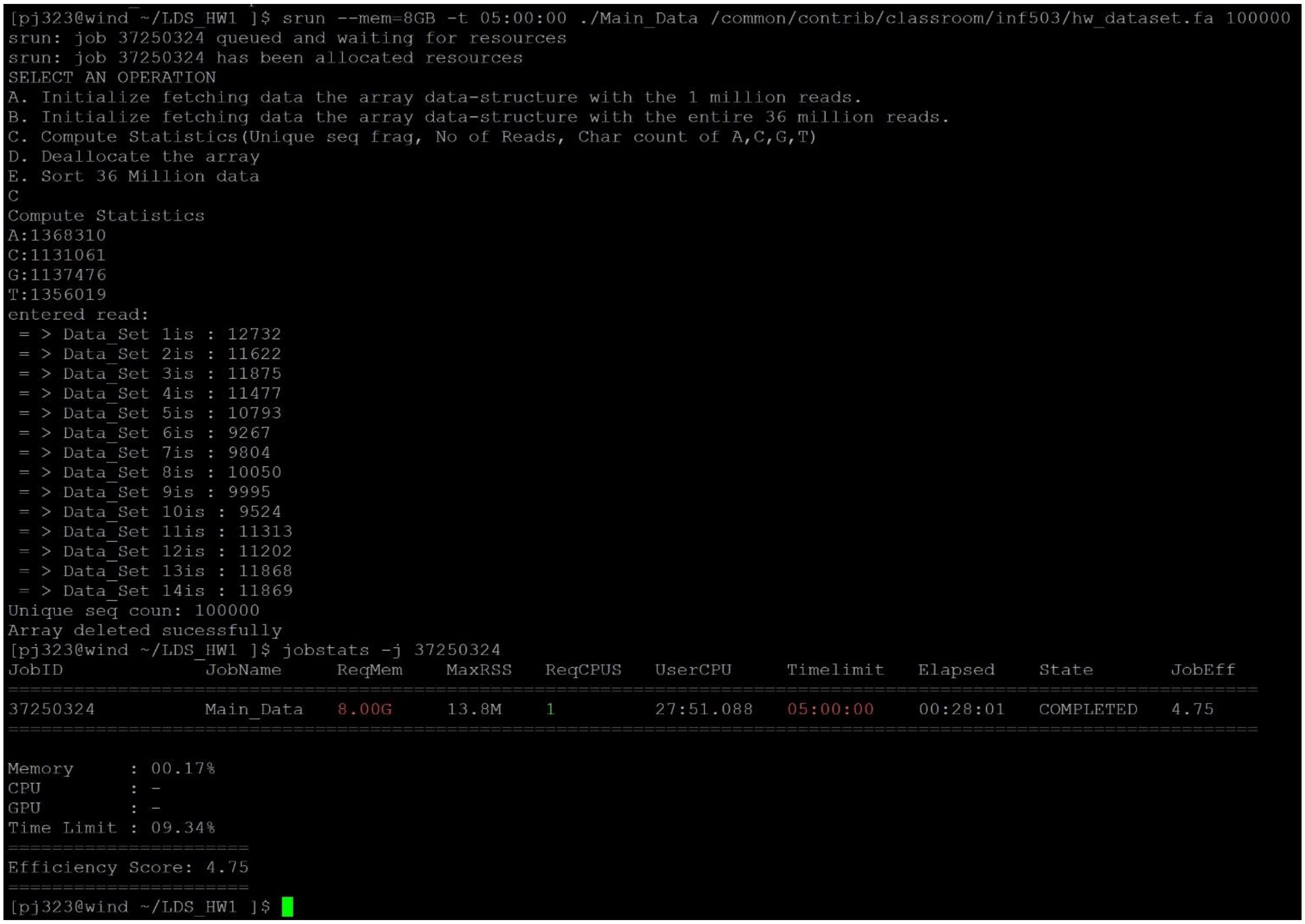
A = 493102372

C = 406639890

G = 408544523

T = 489180159

Computing for **100,000 Data** and checking the unique Sequence so we are calculating all the reads which are unique.



```
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 05:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 100000
srun: job 37250324 queued and waiting for resources
srun: job 37250324 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
C
Compute Statistics
A:1368310
C:1131061
G:1137476
T:1356019
entered read:
 = > Data_Set 1is : 12732
 = > Data_Set 2is : 11622
 = > Data_Set 3is : 11875
 = > Data_Set 4is : 11477
 = > Data_Set 5is : 10793
 = > Data_Set 6is : 9267
 = > Data_Set 7is : 9804
 = > Data_Set 8is : 10050
 = > Data_Set 9is : 9995
 = > Data_Set 10is : 9524
 = > Data_Set 11is : 11313
 = > Data_Set 12is : 11202
 = > Data_Set 13is : 11868
 = > Data_Set 14is : 11869
Unique seq coun: 100000
Array deleted sucessfully
[pj323@wind ~/LDS_HW1 ]$ jobstats -j 37250324
JobID              JobName      ReqMem     MaxRSS    ReqCPUS    UserCPU      Timelimit    Elapsed     State        JobEff
====================================================================================================================
37250324           Main_Data    8.00G      13.8M     1          27:51.088    05:00:00     00:28:01    COMPLETED    4.75
====================================================================================================================

Memory     : 00.17%
CPU        : -
GPU        : -
Time Limit : 09.34%
======================
Efficiency Score: 4.75
======================
[pj323@wind ~/LDS_HW1 ]$ 
```

It copares with every read and chech for the data

Time Taken = **28:01 min**

Ram Used = **13.8 GB**

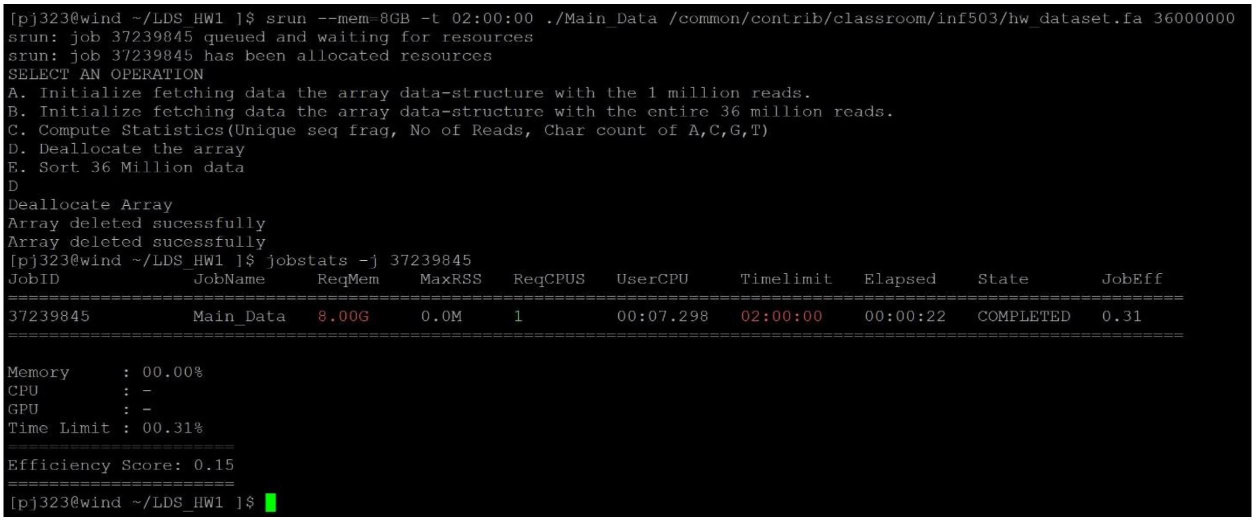
No of Unique Sequence = 100000

Complete computation was done and the Output was given.

Question D

Implement a destructor for your class to delete / deallocate your array data structure. How long did it take? Does this make sense to you?

Basically, we are trying to Deallocate the Array.



```
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 02:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 36000000
srun: job 37239845 queued and waiting for resources
srun: job 37239845 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
D
Deallocate Array
Array deleted sucessfully
Array deleted sucessfully
[pj323@wind ~/LDS_HW1 ]$ jobstats -j 37239845
JobID            JobName     ReqMem    MaxRSS    ReqCPUS    UserCPU      Timelimit    Elapsed     State       JobEff
=========================================================================================================================
37239845         Main_Data   8.00G     0.0M      1          00:07.298    02:00:00     00:00:22    COMPLETED   0.31
=========================================================================================================================

Memory      : 00.00%
CPU         : -
GPU         : -
Time Limit  : 00.31%
======================
Efficiency Score: 0.15
======================
[pj323@wind ~/LDS_HW1 ]$ 
```

~Main_Data()- destructor deallocates the memory of the array as soon as control reaches the end of the class.

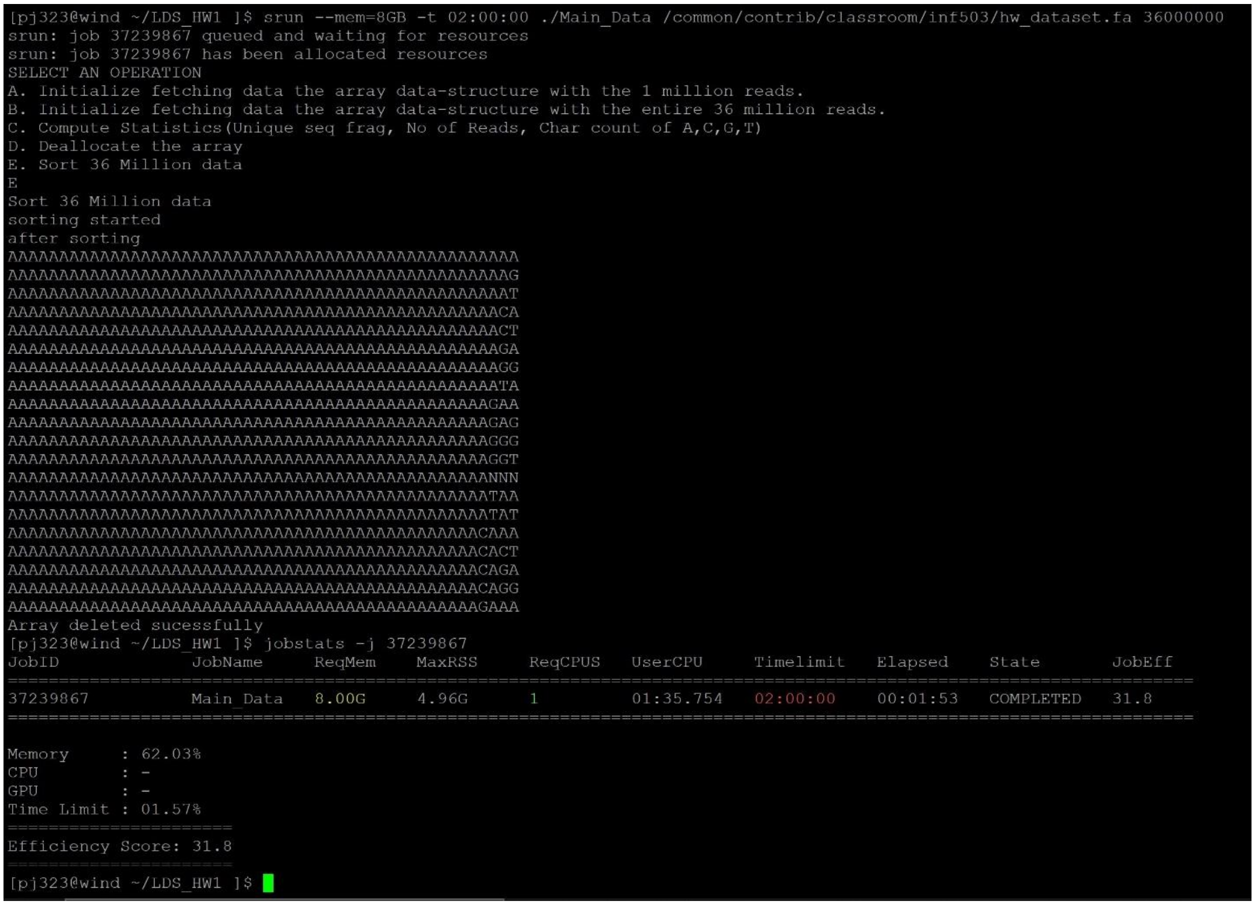
When the function is run on Monsoon it takes around **22 Seconds** to Deallocate the Memory that is present in the Array.

Question E

Implement a function that would sort the genomic sequences (fragments not characters within a fragment) in your array in alphabetic order.

• What is the ‘big O’ notation of your approach (linear / quadratic / cubic / etc)? Please note that depending on the efficiency of your algorithm, you may not be able to alphabetically sort the entire 36 million reads in a reasonable amount of time (24-36 CPU hours). If this happens, reduce the problem size (by using a smaller subset of the reads) and estimate the final run time.

• Print the first 20 lines of the sorted output.



```
[pj323@wind ~/LDS_HW1 ]$ srun --mem=8GB -t 02:00:00 ./Main_Data /common/contrib/classroom/inf503/hw_dataset.fa 36000000
srun: job 37239867 queued and waiting for resources
srun: job 37239867 has been allocated resources
SELECT AN OPERATION
A. Initialize fetching data the array data-structure with the 1 million reads.
B. Initialize fetching data the array data-structure with the entire 36 million reads.
C. Compute Statistics(Unique seq frag, No of Reads, Char count of A,C,G,T)
D. Deallocate the array
E. Sort 36 Million data
E
Sort 36 Million data
sorting started
after sorting
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGGT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAANNN
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATAT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACACT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAGA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAA
Array deleted sucessfully
[pj323@wind ~/LDS_HW1 ]$ jobstats -j 37239867
JobID             JobName      ReqMem    MaxRSS     ReqCPUS    UserCPU     Timelimit   Elapsed     State       JobEff
=======================================================================================================================
37239867          Main_Data    8.00G     4.96G      1          01:35.754   02:00:00    00:01:53    COMPLETED   31.8
=======================================================================================================================

Memory     : 62.03%
CPU        : -
GPU        : -
Time Limit : 01.57%
======================
Efficiency Score: 31.8
======================
[pj323@wind ~/LDS_HW1 ]$ 
```

Time complexity of Merge sort is **O**(nLogn)-which is same in worst case.

It is a logarithmic approach. Space complexity of Merge sort is **O**(n).

It took RAM = **4.96GB** to sort

Time= **1 minute 53 seconds.**