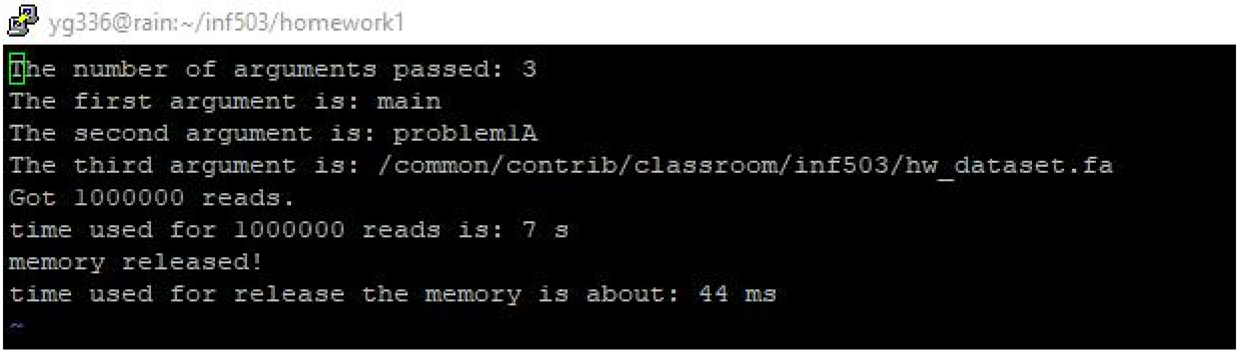
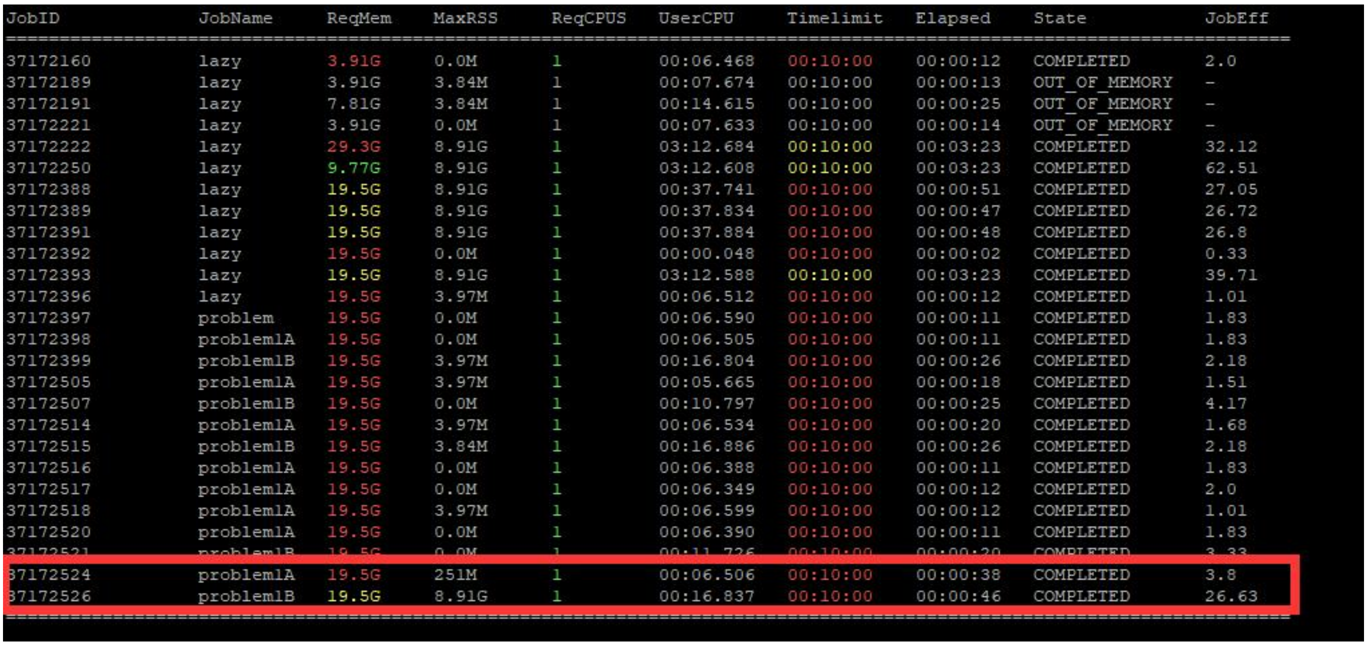
# Question A

For the first 1 million reads, the total CPU time it takes to initialize (fill up) the array is **7 seconds** and the RAM usage is about **251 MB**. For the total 36 million reads, the estimated CPU time is **7*36=252 seconds** and estimated RAM usage is about **251*36=9036 MB**.



```
yg336@rain:~/inf503/homework1
The number of arguments passed: 3
The first argument is: main
The second argument is: problem1A
The third argument is: /common/contrib/classroom/inf503/hw_dataset.fa
Got 1000000 reads.
time used for 1000000 reads is: 7 s
memory released!
time used for release the memory is about: 44 ms
~
```

Fig. 1

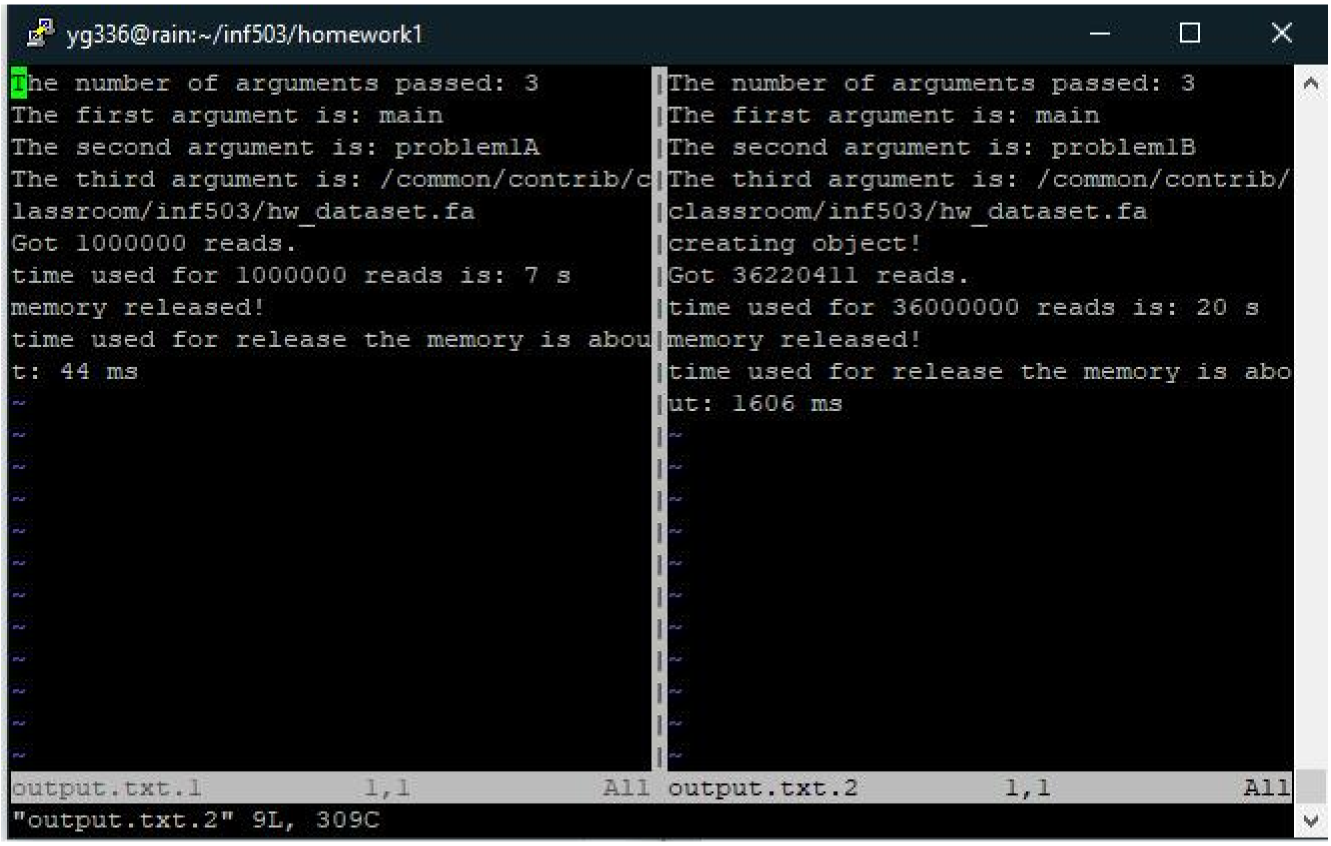


```
JobID           JobName      ReqMem    MaxRSS     ReqCPUS    UserCPU      Timelimit   Elapsed     State           JobEff
=========================================================================================================================
37172160        lazy         3.91G     0.0M       1          00:06.468    00:10:00    00:00:12    COMPLETED       2.0
37172189        lazy         3.91G     3.84M      1          00:07.674    00:10:00    00:00:13    OUT_OF_MEMORY   -
37172191        lazy         7.81G     3.84M      1          00:14.615    00:10:00    00:00:25    OUT_OF_MEMORY   -
37172221        lazy         3.91G     0.0M       1          00:07.633    00:10:00    00:00:14    OUT_OF_MEMORY   -
37172222        lazy         29.3G     8.91G      1          03:12.684    00:10:00    00:03:23    COMPLETED       32.12
37172250        lazy         9.77G     8.91G      1          03:12.608    00:10:00    00:03:23    COMPLETED       62.51
37172388        lazy         19.5G     8.91G      1          00:37.741    00:10:00    00:00:51    COMPLETED       27.05
37172389        lazy         19.5G     8.91G      1          00:37.834    00:10:00    00:00:47    COMPLETED       26.72
37172391        lazy         19.5G     8.91G      1          00:37.884    00:10:00    00:00:48    COMPLETED       26.8
37172392        lazy         19.5G     0.0M       1          00:00.048    00:10:00    00:00:02    COMPLETED       0.33
37172393        lazy         19.5G     8.91G      1          03:12.588    00:10:00    00:03:23    COMPLETED       39.71
37172396        lazy         19.5G     3.97M      1          00:06.512    00:10:00    00:00:12    COMPLETED       1.01
37172397        problem      19.5G     0.0M       1          00:06.590    00:10:00    00:00:11    COMPLETED       1.83
37172398        problem1A    19.5G     0.0M       1          00:06.505    00:10:00    00:00:11    COMPLETED       1.83
37172399        problem1B    19.5G     3.97M      1          00:16.804    00:10:00    00:00:26    COMPLETED       2.18
37172505        problem1A    19.5G     3.97M      1          00:05.665    00:10:00    00:00:18    COMPLETED       1.51
37172507        problem1B    19.5G     0.0M       1          00:10.797    00:10:00    00:00:25    COMPLETED       4.17
37172514        problem1A    19.5G     3.97M      1          00:06.534    00:10:00    00:00:20    COMPLETED       1.68
37172515        problem1B    19.5G     3.84M      1          00:16.886    00:10:00    00:00:26    COMPLETED       2.18
37172516        problem1A    19.5G     0.0M       1          00:06.388    00:10:00    00:00:11    COMPLETED       1.83
37172517        problem1A    19.5G     0.0M       1          00:06.349    00:10:00    00:00:12    COMPLETED       2.0
37172518        problem1A    19.5G     3.97M      1          00:06.599    00:10:00    00:00:12    COMPLETED       1.01
37172520        problem1A    19.5G     0.0M       1          00:06.390    00:10:00    00:00:11    COMPLETED       1.83
37172521        problem1B    19.5G     0.0M       1          00:11.726    00:10:00    00:00:20    COMPLETED       3.33
37172524        problem1A    19.5G     251M       1          00:06.506    00:10:00    00:00:38    COMPLETED       3.8
37172526        problem1B    19.5G     8.91G      1          00:16.837    00:10:00    00:00:46    COMPLETED       26.63
=========================================================================================================================
```

Fig. 2

# Question B

For the entire 36 million reads, the RAM usage is about **8.91 GB** (shown in Question A) and the CPU time is about **20 seconds**.



```
The number of arguments passed: 3           |The number of arguments passed: 3
The first argument is: main                 |The first argument is: main
The second argument is: problem1A           |The second argument is: problem1B
The third argument is: /common/contrib/c    |The third argument is: /common/contrib/
lassroom/inf503/hw_dataset.fa               |classroom/inf503/hw_dataset.fa
Got 1000000 reads.                          |creating object!
time used for 1000000 reads is: 7 s         |Got 36220411 reads.
memory released!                            |time used for 36000000 reads is: 20 s
time used for release the memory is abou    |memory released!
t: 44 ms                                    |time used for release the memory is abo
~                                           |ut: 1606 ms
~                                           |~
~                                           |~
output.txt.1          1,1          All  output.txt.2          1,1          All
"output.txt.2" 9L, 309C
```

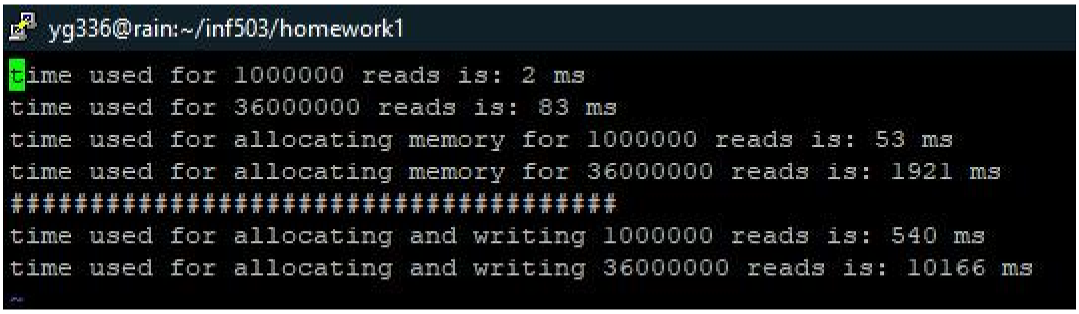
Fig. 3

Compared with the previously estimated time, there is a big gap. In order to understand why there's such a big mismatch, the code in **test.cpp** was used to test.

You may use the following commands to compile and run the code in a Linux system computer:

```
cd /home/yg336/inf503/homework1
g++ -o test.out test.cpp
sbatch test.sh
```

The results are shown in the following figure:



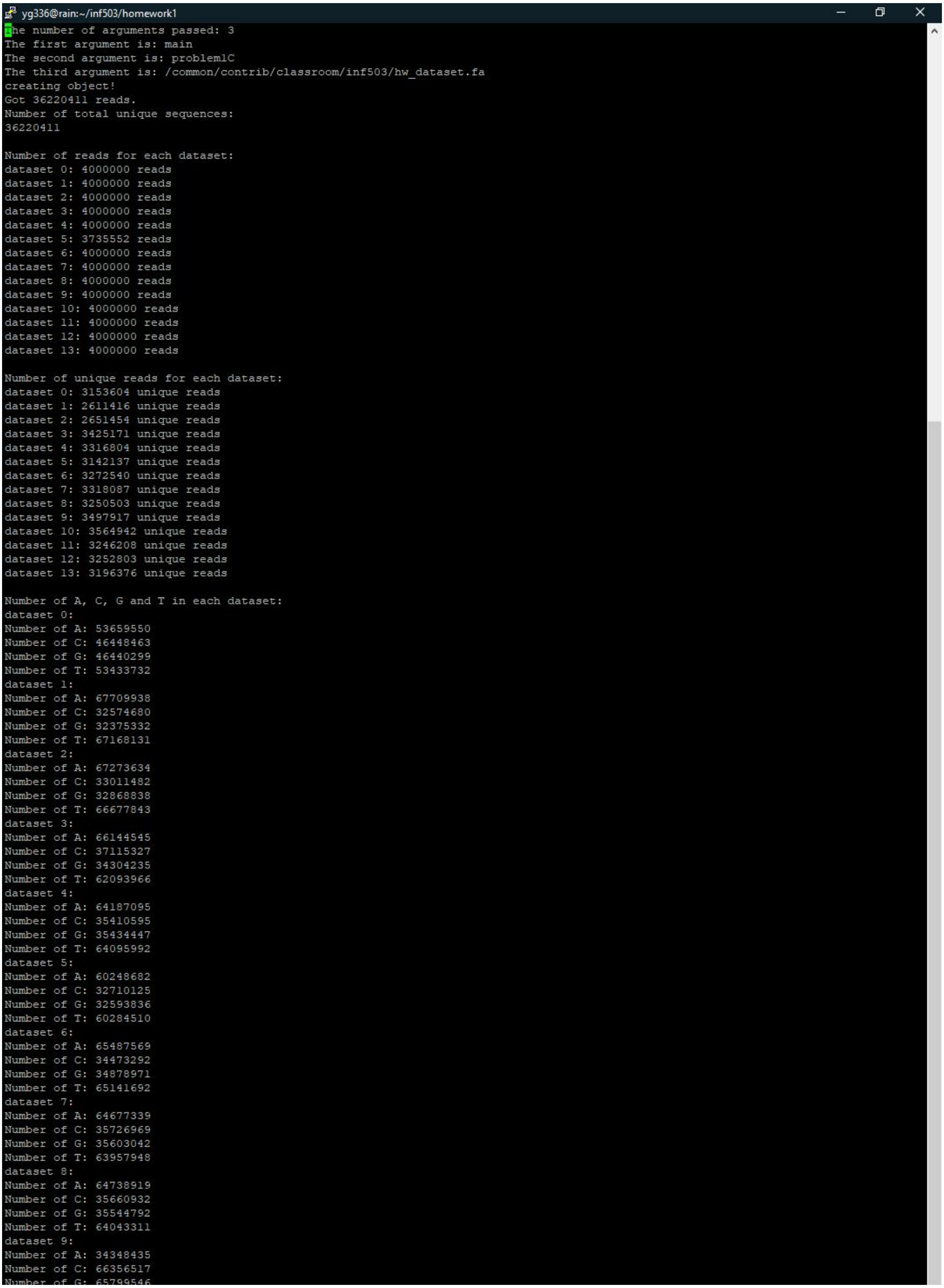
```
time used for 1000000 reads is: 2 ms
time used for 36000000 reads is: 83 ms
time used for allocating memory for 1000000 reads is: 53 ms
time used for allocating memory for 36000000 reads is: 1921 ms
######################################
time used for allocating and writing 1000000 reads is: 540 ms
time used for allocating and writing 36000000 reads is: 10166 ms
~
```

Fig. 4

From the figure we know that if we just allocate memories for the reads, the time consumption will increase linearly. However, if we write data into memories, the time consumption will not increase linearly. Comparing these two processes, we can know that **the time it takes to read data from a file and write to memory is uncertain and that will possibly lead to inaccurate estimation.**

If we compare the results from **test.cpp** to the results shown in Fig. 3, we can get another process that may influence the CPU time consumption: **creating the FASTA_readset object**.

## Question C

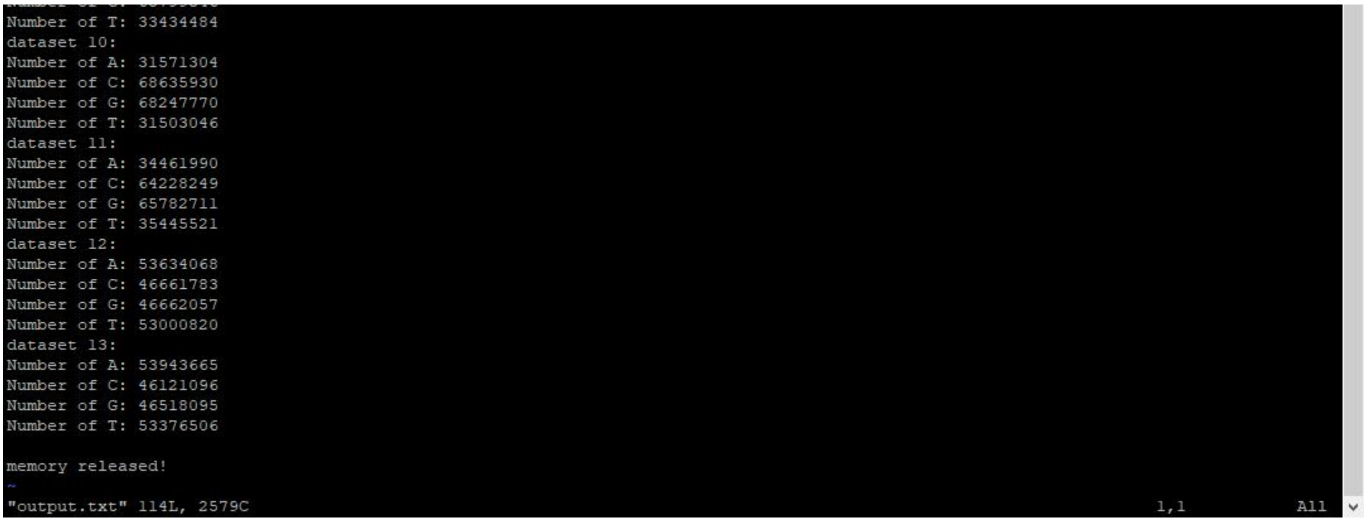


```
The number of arguments passed: 3
The first argument is: main
The second argument is: problem1C
The third argument is: /common/contrib/classroom/inf503/hw_dataset.fa
creating object!
Got 36220411 reads.
Number of total unique sequences:
36220411

Number of reads for each dataset:
dataset 0: 4000000 reads
dataset 1: 4000000 reads
dataset 2: 4000000 reads
dataset 3: 4000000 reads
dataset 4: 4000000 reads
dataset 5: 3735552 reads
dataset 6: 4000000 reads
dataset 7: 4000000 reads
dataset 8: 4000000 reads
dataset 9: 4000000 reads
dataset 10: 4000000 reads
dataset 11: 4000000 reads
dataset 12: 4000000 reads
dataset 13: 4000000 reads

Number of unique reads for each dataset:
dataset 0: 3153604 unique reads
dataset 1: 2611416 unique reads
dataset 2: 2651454 unique reads
dataset 3: 3425171 unique reads
dataset 4: 3316804 unique reads
dataset 5: 3142137 unique reads
dataset 6: 3272540 unique reads
dataset 7: 3318087 unique reads
dataset 8: 3250503 unique reads
dataset 9: 3497917 unique reads
dataset 10: 3564942 unique reads
dataset 11: 3246208 unique reads
dataset 12: 3252803 unique reads
dataset 13: 3196376 unique reads

Number of A, C, G and T in each dataset:
dataset 0:
Number of A: 53659550
Number of C: 46448463
Number of G: 46440299
Number of T: 53433732
dataset 1:
Number of A: 67709938
Number of C: 32574680
Number of G: 32375332
Number of T: 67168131
dataset 2:
Number of A: 67273634
Number of C: 33011482
Number of G: 32868838
Number of T: 66677843
dataset 3:
Number of A: 66144545
Number of C: 37115327
Number of G: 34304235
Number of T: 62093966
dataset 4:
Number of A: 64187095
Number of C: 35410595
Number of G: 35434447
Number of T: 64095992
dataset 5:
Number of A: 60248682
Number of C: 32710125
Number of G: 32593836
Number of T: 60284510
dataset 6:
Number of A: 65487569
Number of C: 34473292
Number of G: 34878971
Number of T: 65141692
dataset 7:
Number of A: 64677339
Number of C: 35726969
Number of G: 35603042
Number of T: 63957948
dataset 8:
Number of A: 64738919
Number of C: 35660932
Number of G: 35544792
Number of T: 64043311
dataset 9:
Number of A: 34348435
Number of C: 66356517
Number of G: 65799546
```

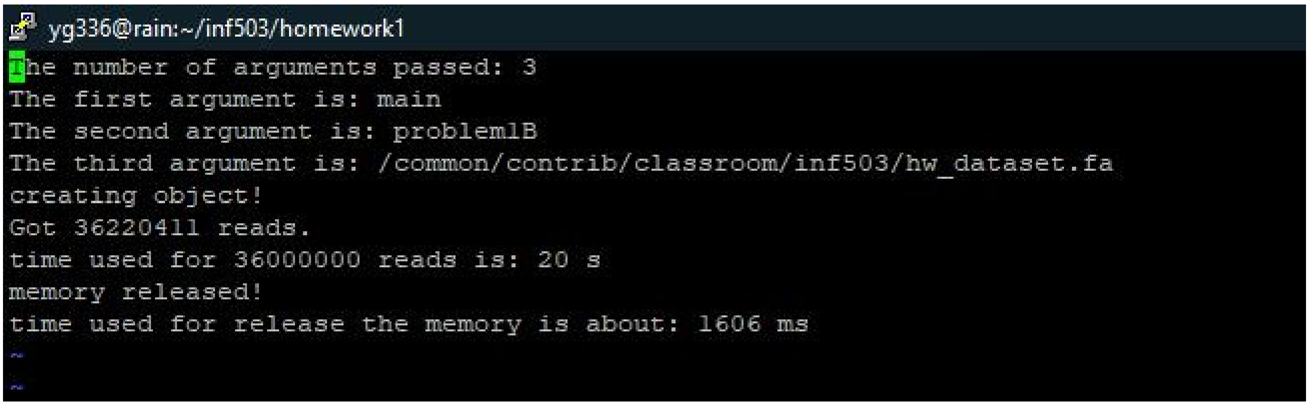


```
Number of T: 33434484
dataset 10:
Number of A: 31571304
Number of C: 68635930
Number of G: 68247770
Number of T: 31503046
dataset 11:
Number of A: 34461990
Number of C: 64228249
Number of G: 65782711
Number of T: 35445521
dataset 12:
Number of A: 53634068
Number of C: 46661783
Number of G: 46662057
Number of T: 53000820
dataset 13:
Number of A: 53943665
Number of C: 46121096
Number of G: 46518095
Number of T: 53376506

memory released!
~
"output.txt" 114L, 2579C                                                        1,1           All
```

Fig. 5

## Question D



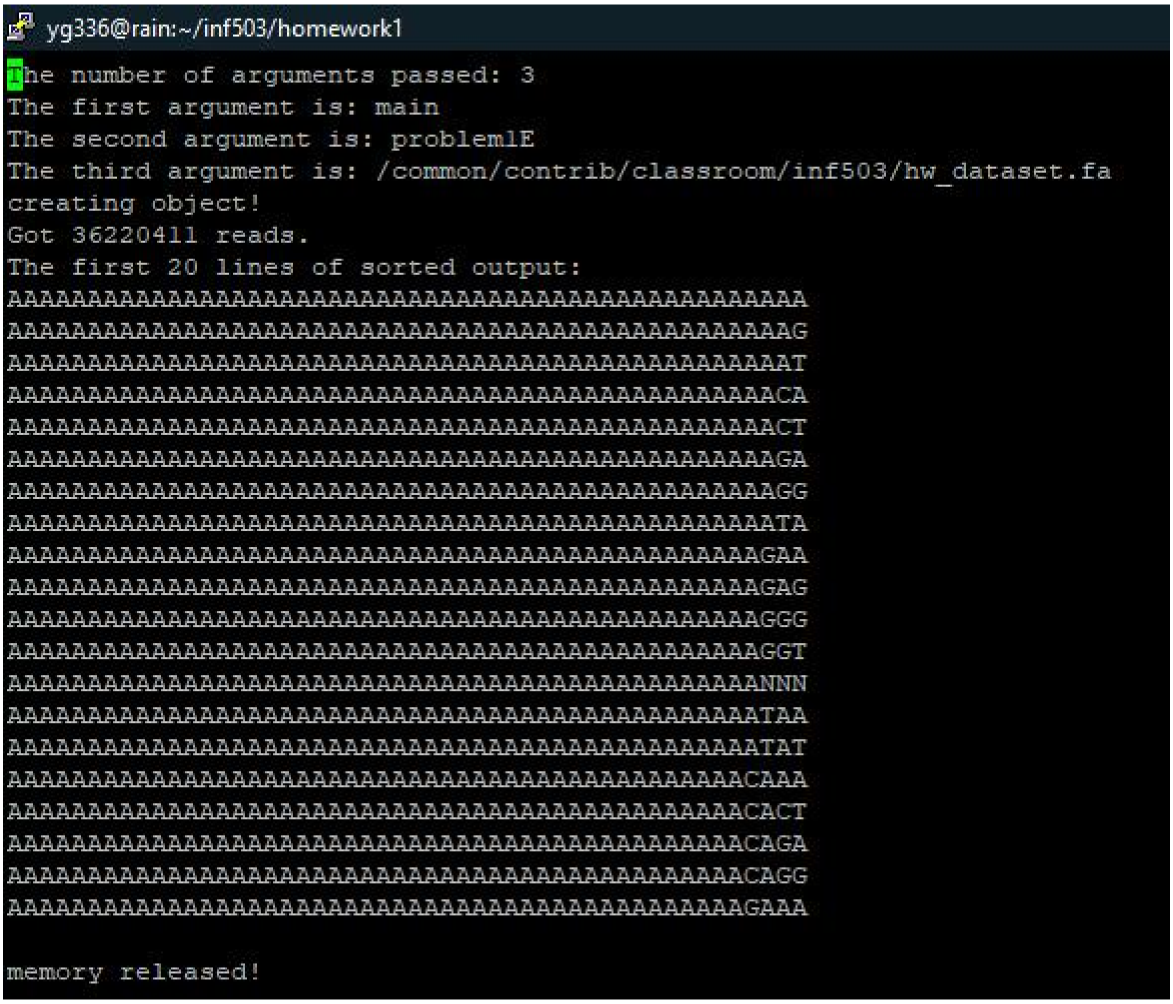
```
yg336@rain:~/inf503/homework1
The number of arguments passed: 3
The first argument is: main
The second argument is: problem1B
The third argument is: /common/contrib/classroom/inf503/hw_dataset.fa
creating object!
Got 36220411 reads.
time used for 36000000 reads is: 20 s
memory released!
time used for release the memory is about: 1606 ms
~
~
```

Fig. 6

From Fig. 6 we can know that it takes **1606 ms** to deallocate the memories. It makes sense. The time increases linearly when comparing with the time shown in Fig. 1.

## Question E

The quick sort was used to sort the segments, so the time complexity is O(nlogn). The first 20 lines are:



```
The number of arguments passed: 3
The first argument is: main
The second argument is: problem1E
The third argument is: /common/contrib/classroom/inf503/hw_dataset.fa
creating object!
Got 36220411 reads.
The first 20 lines of sorted output:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGGT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAANNN
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATAT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACACT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAGA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAA

memory released!
```

Fig. 7