Tyler Thatcher

INF503

HW#1

Homework #1

A. The screenshot below is what jobstats produced for allocating 1 million items in an array. It showed practically no memory being used, not even 1/100 of a GB. I am going to assume that the actual memory is going to be way higher. Something like 36 million / 1000 (MB). My guess is around 36000 bytes or ~360 GB of RAM. As for CPU, I would guess about 10 times the amount of usage, so 00:10.397.

```
JobID           JobName    ReqMem    MaxRSS    ReqCPUS    UserCPU      Timelimit    Elapsed     State        JobEff
=====================================================================================================================
14196764        exercise   234.G     0.00G     1          00:01.397    01:00:00     00:00:09    COMPLETED    0.25
=====================================================================================================================

Requested Memory: 00.00%
Requested Cores : -
Time Limit      : 00.25%
=====================
Efficiency Score: 0.12
=====================
```

B. The screenshot below is what jobstats produced for allocating the entire text file given to us (i.e. 36 million). I expected way more RAM to be used, and I also expected a lot more of the CPU. That being said, I was not accurate at all. I thought it would scale from raw RAM/CPU time. My guess is that C++ does some sort of optimization in the background, which caused this discrepancy.

```
JobID           JobName    ReqMem    MaxRSS       ReqCPUS    UserCPU      Timelimit    Elapsed     State        JobEff
======================================================================================================================
14196773        exercise   234.G     167.G        1          01:43.428    01:00:00     00:03:58    COMPLETED    38.96
======================================================================================================================

Requested Memory: 71.31%
Requested Cores : -
Time Limit      : 06.61%
=====================
Efficiency Score: 38.96
=====================
```

C. The screenshot below shows the stats for Part C.

```
introhw_dataset.fa--C
The filepath is now: /home/tdt62/LargeScaleData/Homework1

------- PART C: A -------
Unique Sequences: 36220412

------- PART C: B -------
Number of reads in dataset 0:  4000000
Number of reads in dataset 1:  4000000
Number of reads in dataset 2:  4000000
Number of reads in dataset 3:  4000000
Number of reads in dataset 4:  4000000
Number of reads in dataset 5:  3735552
Number of reads in dataset 6:  4000000
Number of reads in dataset 7:  4000000
Number of reads in dataset 8:  4000000
Number of reads in dataset 9:  4000000
Number of reads in dataset 10:  4000000
Number of reads in dataset 11:  4000000
Number of reads in dataset 12:  4000000
Number of reads in dataset 13:  4000000
Number of reads in dataset 14:  3801157

------- PART C: C -------
Number of A's: 496119:649
Number of T's: 409131386
Number of C's: 901306047
Number of G's: 411045749
```

D. The total time it took to deallocate my 36 million array was 7 seconds. I assumed it would be fast because it is just deleting memory. Below is an output of my code:

```
Time it took to deallocate: 7 seconds
```

E. I used insertion sort for this problem, for some reason I thought insertion was much faster. The O(N) is O(N^2) which is the same as bubble sort. For this reason, I had to reduce the 36 million to about 10000 to get it to work correctly, and in a timely manner. Below are the screenshots of the sorted array and jobstats of the 10000 items. With the jobstats, it took 4 minutes to store and sort 10000, to do that same thing I would guess it would take around 3600 times longer which is about: which is about 240 hours. This is not doable, and the sorting method should be much faster (merge / quick sort).

- Jobstats

```
[tat62@wind ~/LargeScaleData/Homework1 ]$ jobstats -j 14210210
JobID            JobName    ReqMem     MaxRSS      ReqCPUS    UserCPU      Timelimit    Elapsed     State       JobEff
=================================================================================================================
14210210         exercise   234.G      163.G       1          01:44.871    02:00:00     00:04:07    COMPLETED   36.56
=================================================================================================================

Requested Memory: 69.70%
Requested Cores : -
Time Limit      : 03.43%
======================
Efficiency Score: 36.57
======================
```

- Sorted Array

```
-------- PART E ---------

 -- Sorted Array --
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAATAAAA
AAAAAAAATTGATAACAATATCTTTAGTACTCTACTAGCTAAAGGCTATG
AAAAAAACATCCTGCTTCTGTTGTAAGGCTGTACTTTTACTGGTTAGANN
AAAAAAACCACCATCGCTAACTATATAGGTTCCACCTAGTACGCCACCAG
AAAAAACAACGGCTCAATAAACGACATACATTTCCCGTATGATGACGAGC
AAAAAACAGATCCCTGTAGAAGCCGAACGCCTACAAGAAACTATTCAGTT
AAAAAACAGTTGAAAGTAAGGAGAAGGATGATGGGTTTGATGACTTTGTG
AAAAAACATGTGAAGAATCGCGCACGTTAATCGTGACACTTTGATTATTT
AAAAAACCTGTTGTTATCCCAAGCGGTGTTACGATTAATGTTGCTGCTGG
AAAAAACTTGCCGTCAATGCCGATGCCAGCAAAGTATGCGCTAAGCGATA
AAAAAAGCGGGCATTGACGATGTGGAAACTTCTATGAGCCTCGGTGCGCC
AAAAAAGCTGGAGTAGATATGGCATCTCCAGAACCGGTAAAAGAAGCATT
AAAAAAGGCGGCCGTGTCTTTGTGGAAGATGCACAAGGAAAAGAAATTAA
AAAAAAGGGGGTGAGGGTGTGTAATGATATAACTTTATCTAAAATTCCTT
AAAAAATAAAAAAGTAAAAGGTGCGTATAAAACGCACCTTTTATAATGTA
AAAAAATATTTGCTAATATTAAAAAACTATTGGATCTGGGTTTTTAAAGA
AAAAAATCCCAAAGATAGTCTAAAATTTGTTATCATATTAACTATATTTT
AAAAAATGAAAAACGGCAATCTGCAAGATAGTGCAGAAGTTGTTCGGGAT
AAAAAATTAATTGACTCTGTAAAATAATTAAACACAAATATCTTTATTTA
AAAAAATTTGTACAACAACCAAATATACAAATGGCGGCAGACTATGATCC
AAAAACATAATATTAACTTCTTCAAATAACCACTTCTAACTCTTTTATAA
AAAAACCCAAACACTATCAGGGCACTTGGCTGCCCACCGCCAAGCTGAGC
AAAAACCTGACGTTAGAATATCGCGATCCTTACTGGAACCTCGATGGTGT
AAAAACGATCGCCAGGCCCGCGCGCGCCGTTGGTGACGAACTGCTTGC
AAAAACGGTATTCAGAGCTACCAATAACTCCCCATCCGTATTGCGGAACT
AAAAACTCTACTCTTAGTTTAATCTCTTTGAGTAAATTCTCCGCGATAAC
AAAAACTCTTTTAAAGGTTAATGGCGTAGCATTACCGCCTGCTCCTCCNN
AAAAAGACACGGTTAATCTTCTTGCCGAAATAAGGAATCTTTGATTCACC
AAAAAGACCACCAAAAGGCAGTCCCTCTCAAAACTCAATCCCATTCCCCA
AAAAAGAGATTGACCCAAAAGTAGCTGAATAACTACTTGGGTCAACCACT
AAAAAGATCATTTTTATACGATGTTTTCTATACAACTCTTGTACATCATC
AAAAAGCAGCAACACAGGCTATTCAGGATTTATCAGTTAAANAGGGTGAA
AAAAAGCCGCCCATCTCGACAATATCCCCGCCATGTACGGCATTGTGCGG
AAAAAGGTATCAAAGAGATTTTTGCAACAGTATTCTAGTCTACAAAATTG
AAAAAGTATTTCTATAAGCAAATGGATTGATTTTGGTGGTGGTTGAAATG
AAAAATAAAAAACGTTTAAGAAAACGTTTTCAATAACTTTCATTTATACT
AAAAATAAAGAAAGCTCTTGATCCGCAAAATATTTTGAATCCGAATAAGG
AAAAATAATATTATCACCGCTAAAAGTTTACACAAGATAGCTTTAAAAAG
AAAAATATTTTATTAAAATGAATATTTCTATTCAATTTGGAAAGCATNNN
AAAAATCATAATGTTTTTTTGTAATCAGTATAAAATTGACGTGCTTTAAA
AAAAATCCAAGTCGTATAATAAAATTCCTCGTTTTTGGTCTGATCGGCTT
AAAAATCCGNAAGAGACCCACAAACAAGCTGCCAGCCCCCAAAGCTGGAA
```