Andrew Rutherford

CSCI 3104

CPU: 2.8 GHz Intel Core i7

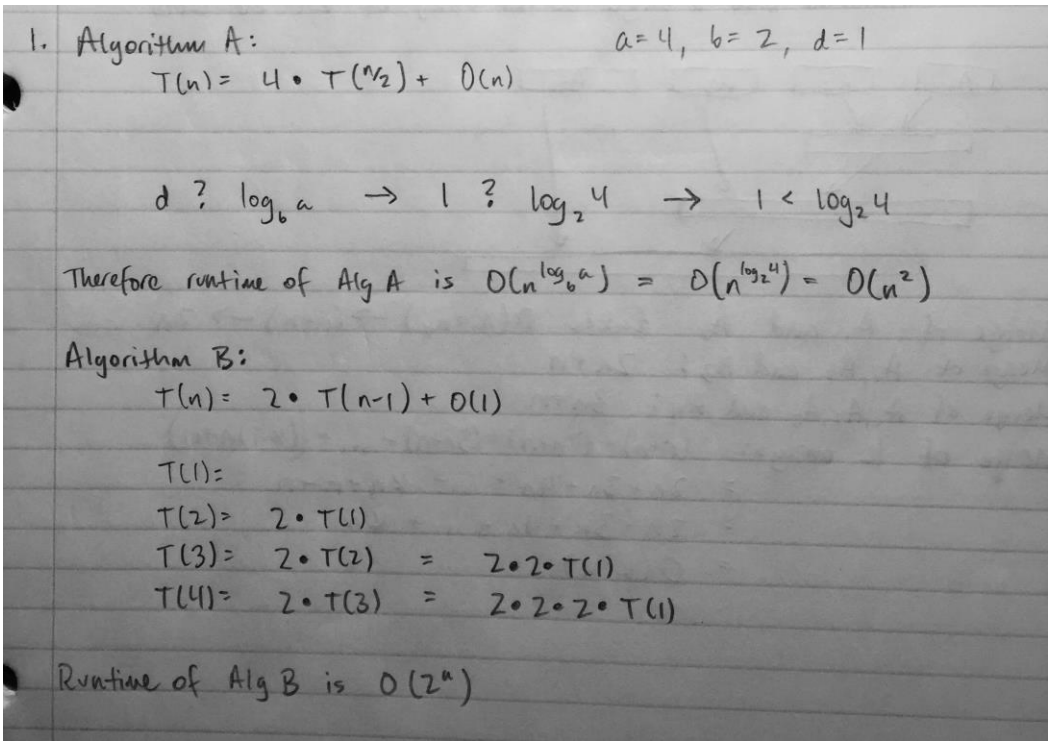
Ram: 16 GB 1600 MHz DDR3

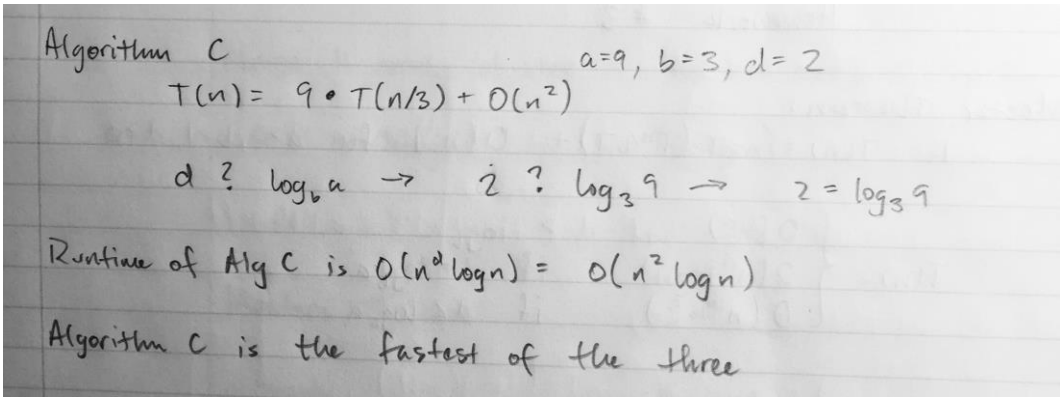
OSX Yosemite

Homework #3

On my honor, as a University of Colorado at Boulder student, I have neither given nor received any unauthorized help.

1.





2.

a. Find the minimum and maximum numbers in the array.

b. Create a new Array A of size M both set at 0.

c. Scan through the array, and for each element X[i], increment A[min{X[i]} + X[i]].

d. Create a new array Y[1 … n].

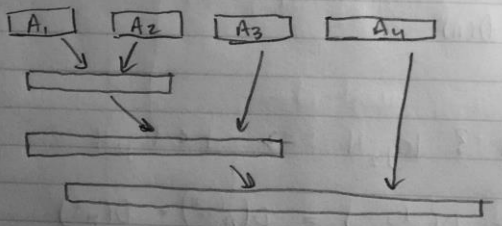
e. Scan through A[] again, for each element A[i], put A[i] values of I into the next empty slots of Y[].

   The output of the sorted array is found by scanning through A[i] and outputting the values in order. It takes a constant amount of work to scan through arrays of size n and M so it takes O(n + M) time.

3.

3. k-way merge sort
   - k sorted arrays, each with n elements
   - combine into a single sorted array of kn elements

a



Merge of A_1 and A_2 takes $O(n_1+n_2) \rightarrow (n+n) \rightarrow 2n$
Merge of A_1, A_2 and A_3: $2n+n$
Merge of A_1, A_2, A_3 and A_4: $3n+n$
Merge of k arrays: $(n+n)+(2n+n)+(3n+n)+\ldots+((k-1)n+n)$

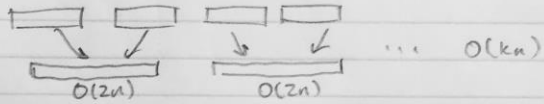
$= 2n+3n+4n+\ldots+kn-n+n$

$= 2n+3n+4n+\ldots+kn$

$= O(k^2n)$

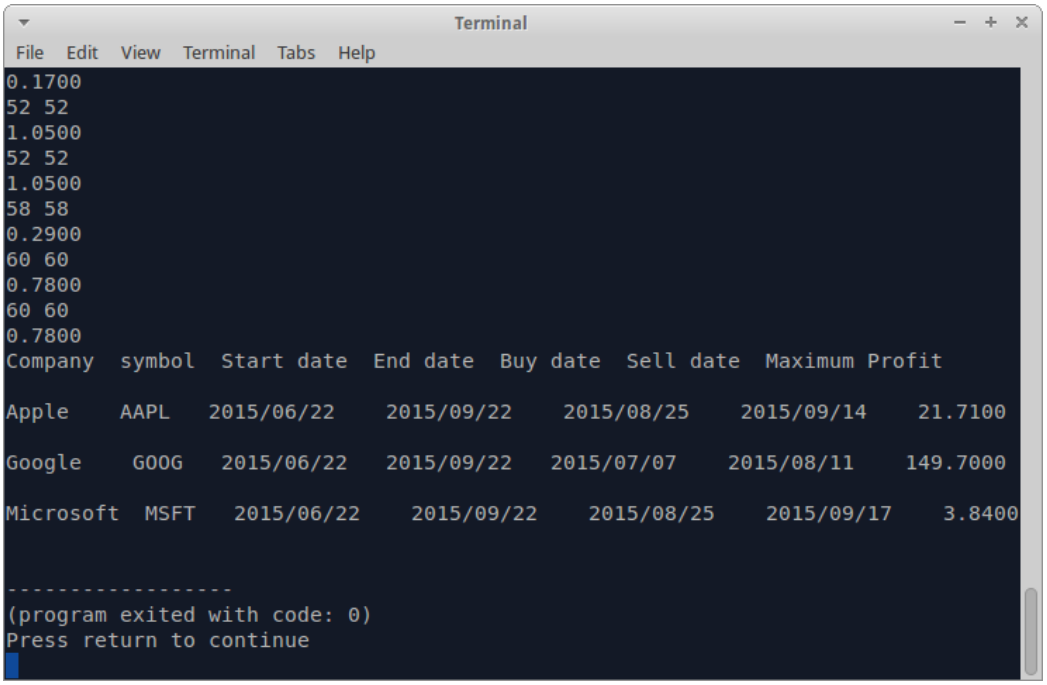
3b Instead of doing it the first way, merge the first array w/ the second, the third array with the fourth, so on for k arrays.

- To merge k arrays of size n into k/2 arrays of size 2n takes $O(kn)$



- This occurs a number of $O(\log k)$ times
- Total run time is $O(\log k) \cdot O(kn) = O(kn \log k)$

4.


Terminal
File Edit View Terminal Tabs Help
0.1700
52 52
1.0500
52 52
1.0500
58 58
0.2900
60 60
0.7800
60 60
0.7800
Company symbol Start date End date Buy date Sell date Maximum Profit
Apple AAPL 2015/06/22 2015/09/22 2015/08/25 2015/09/14 21.7100
Google GOOG 2015/06/22 2015/09/22 2015/07/07 2015/08/11 149.7000
Microsoft MSFT 2015/06/22 2015/09/22 2015/08/25 2015/09/17 3.8400
(program exited with code: 0)
Press return to continue