***Kostaq Papa***

***0412768***

***CS2134***

***HW01***

***Due September 20, 2011; 11:59PM***

**Written Part:**

Problem 1) For each of the following functions, determine the Big-Oh notation:

1. O(n)
2. O(log(n))
3. O(log (n))
4. O(n3)
5. O(n)
6. O(n log3(n))
7. O(n log(n))
8. O(n2)

Problem 2) For each of the following code fragments, determine the worst case running

time using Big-Oh notation, as a function of n.

1. O(n)
2. O(log(n))
3. O(n2)
4. O(log(n))
5. O(n log(n))

Problem 3) Suppose a function takes 0.0005 seconds to run on input size of 512: Give an estimate on how long its running time would be for an input size of 2048 if:

1. the function had an O(n) running time.

For n=512, time=0.0005s.

Therefore, for n=2048, time=0.002s

1. the function had an O(n4) running time.

For n=512, time=0.0005s.

If O(n4), therefore, for n=2048, time = = 0.21147

1. the function had an O(n2) running time.

For n=512, time=0.0005s.

If O(n2), therefore, for n=2048, time = = 0.04472

Problem 4) Arrange the following in increasing order:

O(n log(n)), O(n2 log(n)), O((log(n))2), O(n), O(n2), O(n3), O(2n)

Problem 5)

a) No, the time will not greatly vary. The Big O will remain the same even if different numbers were assigned.

b) If the vector was unsorted and a sequential search is used, one run of the problem would greatly vary from the other. But if the vector was sorted and binary search was used one problem run wouldn’t greatly vary from the other.

Problem 6)

Look at the output from running the programming part.

Did maxSubsequenceSum1 have O(n3) time?

Yes, the maxSubsequenceSum1 has O(n3) time.

Did maxSubsequenceSum2 have O(n2) time?

Yes, the maxSubsequenceSum2 has O(n2) time.

Did maxSubsequenceSum4 have O(n) time?

Yes, the maxSubsequenceSum4 has O(n) time.

Predict (roughly) how long each algorithm would take if n = 215. Show how you made your prediction.

For maxSubsequenceSum1, Big O is O(n3).

For n=245, the time was 0.042s. Therefore, for n=215, time=(215\*0.042)/245 = 0.039s

For maxSubsequenceSum2, Big O is O(n2).

For n=245, the time was 0.001s. Therefore, for n=215, time=(215\*0.001)/245 = 0.000877s

For maxSubsequenceSum4, Big O is O(n).

The running time was 0 for all the required n values. Therefore I am predicting that for n=215 the running time will be 0 too.