There are 50 points.                                    Time limit: 2.5 hours.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part I. Please select one best answer. There are 10 questions. Each is worth one point. Make sure you circle exactly one answer.**

(1) If f(n) is Ө(g(n)) then

(A)  g(n) is not Ω(f(n))

(B)  f(n) is not Ω(g(n))

(C)  f(n) is not O(g(n))

(D) g(n) is not o(f(n))

(E)   All of the above

(2) A problem Q is in class NP means

(A)  Q has a non-polynomial time complexity.

(B)  Q does not have a polynomial time algorithm.

(C)  All known algorithms of Q have an exponential time complexity.

(D) Q can be verified in polynomial time.

(E)   None of the above.

(3) What is the recurrence relation to compute the time complexity of merge sort?

(A)  T(n) = T(n/2) + cn

(B)  T(n) = 2T(n/2) + cn2

(C)  T(n) = T(n/2) + cn2

(D) T(n) = 2T(n/2) + c

(E)   None of the above.

(4) Miss Marple has 5 Red apples, 3 Green apples and 2 Yellow apples in her fruit basket. She picks an apple, notes down its color, and then puts it back in the basket. On average, we expect Miss Marple to pick \_\_\_\_\_ Red apples, before she can pick 15 Green apples.

(A)  10

(B)  15

(C)  25

(D)  35

(E)   None of the above.

(5) What is log162?

(A)  2

(B)  4

(C)  0.25

(D) 0.5

(E)   None of the above

(6) Assuming base =2, if log(log n) =4, what is n?

(A)  512

(B)  256

(C)  128

(D) 64

(E)   None of the above

(7) Consider the following code. What is its time complexity?

int questionOne(int n)

{

       int count = 0;

       for (int i=n;  i>0; i/=2)

                  for (int j=i-1; j>= 0; j--)

                            count += 1;

       return count;

}

(A)  Ө(n)

(B)  Ө(log n)

(C)  Ө(n log n)

(D) Ө(n2)

(E)   None of the above.

(8) Consider the following code. What is the time complexity?

int questionTwo(int n)

{

       int count = 0;

       for (int i=0;  i<n; i++)

                  for (int j=n; j>1; j--)

                            count = count + 1;

       return count;

}

(A)  Ө(n)

(B)  Ө(log n)

(C)  Ө(n log n)

(D) Ө(n2)

(E)   None of the above.

(9) Let W(n) and A(n) denoted respectively, the worst case and the worst case running time of an algorithm executed on an input of size n. Which of the following is **always true**?

(A)  A(n) is in Ω(W(n))

(B)  A(n) is in Ө(W(n))

(C)  A(n) is in O(W(n))

(D) A(n) is in o(W(n))

(E)   None of the above.

(10) Which of the following is an **incorrect** statement?

(A)   Time complexity of n × n chess is exponential.

(B)  Average case Time complexity of QuickSelect is O(n).

(C)  Halting problem is an example of a problem having an algorithmic solution but no definable solution.

(D)  All feasible problems have at least one algorithmic solution with time complexity O(nk) for some non-negative positive integer k.

(E)  None of the above.

**Part II. Short answer. 2 points each. There are 12 questions.**

(11)  If you know log(f(n)) is Ө(log(g(n)), can you conclude f(n) is Ө(g(n))? Show the logical reasoning for your answer. It can be through examples.

(12)  What is the lower bound for all inversion bound sorting algorithms? Show the logical reasoning for your answer. It **cannot** be through examples.

(13) What is the lower bound of all comparison-based sorting algorithms? Show the logical reasoning for your answer. It **cannot** be through examples.

(14) Name two sorting algorithms that are comparison-based but not inversion bound. You must show the algorithms you are listing are not inversion bound.

(15) What is the sum of the finite series 1/2 + 1/2 +3/8 + 1/4 + 5/32 + … ?

(16)  Proof by induction the result

1/(1 × 2) + 1/(2 × 3) + ... + 1/(n × (n+1)) = n/(n+1)

(17) Not captured.

(18) Can you think of a binary search algorithm for searching an item in a linked list?

If such an algorithm is possible, write the algorithm.

If such an algorithm doesn’t make sense, explain your reasoning.

(19) Draw the decision tree for sorting three data items: a, b and c as in your class notes.

(20) Not captured.

(21) Not captured.

 (22) What is the time complexity of Radix Sort?

If you are using variable names such as m, n, k and so on, you must clearly specify what they mean in your expression.

**Part III. There are four questions of varying points in this part.**

(23) [4 points] in the class, Prof. has shown you how he got 7 when the array is resized by doubling. Show the corresponding calculation if the array is resized 8 times the current size.

[1 point] What is the value of c(resize)?

[1 point] What is the value c(add)?

[2 points] How did you get the value of c(add)? Explain clearly.

(24) [3 points] Sort the list of numbers 67,39, 27, 79, 59, 51, 3 using Radix sort. Radix = 8. ( Just writing in sorted order will get you 0. You must show all the steps.)

(25) [3 points] Illustrate QuickSelect({69, 29, 19, 79, 49,39,59}, 5). Pick the pivot using the ***median of three*** ***rule***. You must show all the steps. Just writing the correct answer **59** is worth **0** points. You must use integer division as we did in the class all the time.

(26) [6 points] Consider the following algorithm.

Algorithm Happy(int n)

if(n==0)

return

else

Happy(n-1)

println(“I am quite happy today!”)

Happy(n-1)

end if

1. [1 point] What is the output for n=3?
2. [1 point] Write the recurrence relation to compute T(n).
3. [2 points] Using your answer to part (b), determine T(n).
4. [1 point] From your answer to part ©, what is the time complexity of Algorithm Happy?
5. [1 point] Is this a feasible solution? Please answer “Yes” or “No”. Justify your answer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_