Analysis and Design of Algorithms
Tutorial Assignment No 1
Introduction and Review

Date of submission : Will be specified two days before the due date

Date uploaded :8th August 2019

Instructions:

- *You are supposed to write the answers to each question in a notebook and submit the same on the day you are asked to submit, without fail.*
- In all those problems where you are asked to write the algorithm, it is necessary that you write the pseudocode as per the conventions discussed in the class - the conventions used in the CLRS text, analyze the algorithm using mathematical analysis or asymptotic analysis as specied and give the appropriate expression for the time complexity
- *The late submissions would incur a penalty of 5 marks per day, cumulated.*
- The plagiarism would be dealt with as discussed in the class, without any arguments.
- *Be precise and succinct in answering the questions. Unnecessary elaboration will not yield any extra bonus.*

1. Give at least two different real-life examples of a subjective decision and a non-subjective decision.

2. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To find the sum of n elements in an integer array without using recursion.

3. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To perform the bubble sort.

4. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To find the smallest element from an integer array.

5. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To find the factorial of a given number without using recursion.

6. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To find the n^{th} Fibonacci number without using recursion.

7. Devise the algorithm and perform the mathematical analysis as illustrated in the class for the following: To sort an input array of n integers using insertion sort.

8. Create a data set or find a dataset from the internet - consisting of at least a million interger values in a vector. Write the Insertion sort routine in C and time the function to sort using the structure *timeval* as discussed in class.

```
typedef struct timeval {long tv_sec;
long tv_usec;
} timeval;
```

   Now repeat the same on the sorted output. Note the difference in time - in sorting an unsorted interger vector and a sorted one.

9. Repeat the above exercise for the Bubble sort, the Merge sort and the Quick sort covered. Note the time differences.

10. Do a bit of research and find out on what factors does the actual time taken by a program depends on. Use these factors as an argument to justify why it is not feasible to do empirical analysis when the goal is to compare two algorithms.

11. Devise an algorithm to find the sum of elements of a vector consisting on n elements, recursively. Devise the recurrence relation for the same and solve it, to find out the time complexity.

12. What is/could be the input size, in the following ?

    - Find x in an array of names
    - Multiply two matrices with real entities
    - Sort an array of numbers
    - Traverse a binary tree
    - Solve a problem concerning graphs

13. Design the recursive version of the Fibonacci algorithm and only obtain the recurrence relation.

14. Design an algorithm for matrix addition and analyze its time complexity.

15. Find the cost of execution of the following code snippet

```
for i = 1 to n
        for j = 1 to i
                x = x + 1
```

16. Find the cost of execution of the following code snippet

```
j=n
while (j >= 1){
        for i = 1 to j
                x = x + 1
        j = n/2
}
```

17. Find the cost of execution of the following code snippet

```
for i = 1 to n
        for j = 1 to i
                for k = 1 to i
                        x = x + 1
```

18. Find the cost of execution of the following code snippet

```
i=n
while (i >= 1){
        for j = 1 to n
                x = x + 1
        i = i/2
}
```

19. Write an algorithm EXPONENT(a, n) to find an using an appropriate method. Analyze the asymptotic complexity of the algorithm and compare it with the conventional method to do so.

20. IF $f(n) = 100 * 2^n + 8n^2$, prove that $f(n) = O(2^n)$. Can you claim that $f(n) = \theta(2^n)$. IF so, prove the same.

21. Is it correct to say that $f(n) = 3n+8 = \Omega(1)$?. Given the facts that f(n)=$3n+3=\Omega(n)$ and f(n)=$3n+3 = \Omega(1)$, which one is correct ? Which one would you choose to prescribe the growth rate of f(n) ?

22. Consider the two functions viz. $f(n) = n^2$ & $g(n) = 2n^2$. Which functions growth rate is higher ? USe appropriate asymptotic notation to specify the time complexity of the two functions.

23. Prove the following: *For any two functions f(n) and g(n), $f(n) = \theta(g(n))$ only if f(n)=O(g(n) and f(n)=$\Omega(g(n))$.*

24. Solve the following problems:

    a. Show that $T(n) = 1 + 2 + 3 + ..n = \Theta(n^2)$

    b. Prove that $2n^3n^2 = O(n^3)$

    c. Prove that $7n^2logn + 25000n = O(n^2logn)$

25. If $T1(n) = O(f(n))$ and $T2(n) = O(g(n))$ then show that (a) $T1(n) + T2(n) = max(O(g(n), O(f(n))$ (b) $T1(n) * T2(n) = O((g(n) * (f(n))$

26. Show that $maxf(n), g(n) = \Theta(f(n) + g(n))$

27. Prove or disprove: (a) $n^2 2^n + n^{100} = (n^2 2^n)$ (b) $n^2/logn = \Theta(n^2)$

28. Prove that if T(x) is a polynomial of degree n, then $T(x) = \Theta(x^n)$.

29. If P(n) is any polynomial of degree m or less then show that $P(n) = a^0 + a_1 n + a_2 n^2 + + a_m n^m$ then $P(n) = O(n^m)$

30. Find the running time of the following algorithm:

```
Algorithm SUM(n)
1.      answer = 0;
2.      for i= 1 to n do
3.          for j= 1 to i do
4.                for k = 1 to j do
5.                 answer++;
6.      print(answer);
```

31. Let A and B be two programs that perform the same task. Let $t_{A(n)}$ and $t_{B(n)}$ respectively denote their values. For each of the following pairs, find the range of n value for which program A is faster than program B :

    a. $t_{A(n)} = 1000n$ and $t_{B(n)} = 10n^2$

    b. $t_{A(n)} = 1000nlog_2 n$ and $t_{B(n)} = n^2$

    c. $t_{A(n)} = 2n^2$ and $t_{B(n)} = n^3$

    d. $t_{A(n)} = 2n$ and $t_{B(n)} = 100n$

32. Consider an input array A of n elements. Each element is an n-bit integer except 0. Which sorting algorithm would you recommend for sorting the array ? Why ? What will be the complexity your sorting algorithm ? [Hint: What is the range in which each array value (i.e. a number) i.e. an integer falls into ?]

33. Given the following statement viz. *Consider an input array $a[1..n]$ of arbitrary numbers. It is given that the array has only $O(1)$ distinct elements. What does the statement imply ?*