| Faculty of Engineering and Technology | | | |
|---|---|---|---|
| Ramaiah University of Applied Sciences | | | |
| **Department** | Computer Science and Engineering | **Programme** | B Tech. in CSE |
| **Semester/Batch** | 4th /2018 | | |
| **Course Code** | 19CSE214A | **Course Title** | Design and Analysis of Algorithms |
| **Course Leader** | Vaishali R Kulkarni / Pallavi R Kumar | | |

## Assignment

| Reg.No. | | Name of Student | |
|---|---|---|---|

| Sections | | Marking Scheme | Max Marks | First Examiner Marks | Moderator |
|---|---|---|---|---|---|
| **Part A** | **A 1.1** | **Introduction** | 2 | | |
| | **A 1.2** | **Set of case studies** | 2 | | |
| | **A 1.5** | **Conclusion** | 1 | | |
| | | **Part-A  Max Marks** | 05 | | |
| **Part B** | **B 1.1** | **Pseudocode for each approach** | 3 | | |
| | **B 1.2** | **Discussion on efficiency of algorithms** | 4 | | |
| | **B 1.3** | **Results of three approaches using C programming** | 3 | | |
| | | **B.1 Max Marks** | 10 | | |
| | **B 2.1** | **Pseudocode for each approach** | 3 | | |
| | **B 2.2** | **Discussion on efficiency of algorithms.** | 3 | | |
| | **B 2.3** | **Comparison of time/ space complexity for each algorithm.** | 4 | | |
| | | **B.2 Max Marks** | 10 | | |
| | | **Total Assignment Marks** | 25 | | |

## Course Marks Tabulation

| Component-1 (B) Assignment | First Examiner | Remarks | Moderator | Remarks |
|---|---|---|---|---|
| A | | | | |
| B.1 | | | | |
| B.2 | | | | |
| Marks (out of 25 ) | | | | |

**Signature of First Examiner**                    **Signature of Moderator**

**Please note:**

1. Documental evidence for all the components/parts of the assessment such as the reports, photographs, laboratory exam / tool tests are required to be attached to the assignment report in a proper order.

2. The First Examiner is required to mark the comments in RED ink and the Second Examiner's comments should be in GREEN ink.

3. The marks for all the questions of the assignment have to be written only in the **Component – CET B: Assignment** table.

4. If the variation between the marks awarded by the first examiner and the second examiner lies within +/- 3 marks, then the marks allotted by the first examiner is considered to be final. If the variation is more than +/- 3 marks then both the examiners should resolve the issue in consultation with the Chairman BoE.

## Assignment

**Instructions to students:**

1. The assignment consists of **3** questions: Part A-**1** Question, Part B-**2** Questions.

2. Maximum marks is **25**.

3. The assignment has to be neatly word processed as per the prescribed format.

4. The maximum number of pages should be restricted to **10**.

5. Restrict your report for Part-A to 1 page only.

6. Restrict your report for Part-B to a maximum of 9 pages.

7. The printed assignment must be submitted to the course leader.

8. **Submission Date: March 20, 2020**

9. **Submission after the due date is not permitted.**

10. **IMPORTANT**: It is essential that all the sources used in preparation of the assignment must be suitably referenced in the text.

11. Marks will be awarded only to the sections and subsections clearly indicated as per the problem statement/exercise/question.

**Course Preamble**

This course is intended to teach the principles and concepts of design and analysis of computer algorithms. Formal proofs of algorithms and techniques for analyzing their best case, worst case, average case and amortized complexities are covered. Brute force, greedy, divide-and-conquer, branch-and-bound, backtracking and dynamic programming based algorithm design techniques are covered in detail and illustrated using examples. Complexity classes, approximations, heuristics and randomized algorithm techniques are discussed. Students are trained to design and analyze algorithms for the given computational problems using appropriate techniques and methods.

Divide and Conquer is an algorithmic pattern. In algorithmic methods, the design is to take a dispute on a huge input, break the input into minor pieces, decide the problem on each of the small pieces, and then merge the piecewise solutions into a global solution. This mechanism of solving the problem is called the Divide & Conquer Strategy.

In this context write a review on "***Design of Divide and Conquer algorithms***"

Your report should include the following:

        **A .1.1** Introduction

        **A .1.2** Set of case studies

        **A .1.3** Conclusion

**Part B**                                                                                                            **(20 Marks)**

**B .1**                                                                                                            (10 Marks)

Consider a travelling salesman problem, where a salesman travels a set of cities with the shortest possible route that visits every city exactly once and returns to the starting point. Design algorithms to solve the travelling salesman problem using greedy algorithm, dynamic programming and backtracking approach. Compare the cost of travelling salesman problem tour by each method and also compute the actual execution time. Your report should include:

        B.2.1 Pseudocode for each approach

        B.2.2 Discussion on efficiency of algorithms

        B.2.3 Results of three approaches using C programming

B .2                                                                                                            **(10 Marks)**

Consider a list of $n-1$ integers and these integers are in the range of 1 to $n$. There are no duplicates in the list. One of the integers is missing in the list. Design an algorithm to find the missing integer.
Example: Input: [1, 2, 4, 6, 3, 7, 8] , Output: 5.
Design three algorithms based on the Brute Force, Sorting and Hashing approach. Implement algorithms using C programs. Your report should include the following:

        **B.2.1** Pseudo code to find a missing number using the above-mentioned approaches.

        **B.2.2** Discussion of efficient algorithm to find a missing number.

        **B.2.3** Comparison of time and space complexity for each algorithm.