**Department of Computer Science**

**SECOND SEMESTER M.Sc. DEGREE EXAMINATION, APRIL 2023**

**SCHEME OF VALUATION**

**CC19P CSS2 C06 - DESIGN AND ANALYSIS OF ALGORITHMS 22P258**

**Max weightage: 30**

**Faculty Name: Ms.Viji Viswanathan**

**PART A**

**Answer any four questions. Each question carries 2 weightage.**

|  |  |  |  |
| --- | --- | --- | --- |
| Question No | Marks/Grades allotment | Marks/Grades |  |
| 1  Ans. | Identify the Four Major Stages of Algorithm Analysis and Design. |  |  |
| Design  Analyze  Implement  Experiment  (Only stages) | C | 4 |
| Only explanation without distinct stages | A | 8 |
| Stages + Explanation | A+ | 10 |
| 2  Ans | Illustrate divide and conquer method. |  |  |
| Divide And Conquer  This technique can be divided into the following three parts:  Divide: This involves dividing the problem into smaller sub-problems.  Conquer: Solve sub-problems by calling recursively until solved.  Combine: Combine the sub-problems to get the final solution of the whole problem. | B | 6 |
| Explanation with one problem | A | 8 |
| Explanation with two or more problems. | A+ | 10 |
| . 3  Ans. | Illustrate knapsack problem |  |  |
| We are given N items where each item has some weight and profit associated with it. We are also given a bag with capacity W, [i.e., the bag can hold at most W weight in it]. The target is to put the items into the bag such that the sum of profits associated with them is the maximum possible. | A | 8 |
| Explanation with example | A+ | 10 |
| 4  Ans. | Criticize time and space complexity It is a strategy |  |  |
| Time Complexity: The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.  Space Complexity:  Definition –  Problem-solving using computer requires memory to hold temporary data or final result while the program is in execution. The amount of memory required by the algorithm to solve given problem is called space complexity of the algorithm.. | A | 8 |
| Definition + example | A+ | 10 |
| 5  Ans | Debate the difference between big oh &big omega notation |  |  |
| 1. Big-oh notation: Big-oh is the formal method of expressing the upper bound of an algorithm's running time.  f (n) ⩽ k.g (n)f(n)⩽k.g(n) for n>n0n>n0 in all case  graph  2. Omega () Notation: The function f (n) = Ω (g (n)) [read as "f of n is omega of g of n"] if and  only if there exists positive constant c and n0 such that  F (n) ≥ k\* g (n) for all n, n≥ n0 | B | 6 |
| Definition+graph+example | A+ | 10 |
| 6 | Describe clique |  |  |
| Ans. | In Clique, every vertex is directly connected to another vertex, and the number of vertices in the Clique represents the Size of Clique. | B | 6 |
|  | To Prove: - Clique is an NPC or not?  For this you have to satisfy the following below-mentioned points: -  1. Clique  2. 3CNF ≤ρ Clique  3. Clique ≤ρ 3CNF≤SAT  4. Clique ϵ NP | A+ | 10 |
| 7 | Analyse parallel computing. Why do we use it? |  |  |
| Ans. | Parallel computing is a type of computing architecture in which several processors simultaneously execute multiple, smaller calculations broken down from an overall larger, complex problem. Parallel computing is a type of computing architecture in which several processors simultaneously execute The whole real-world runs in dynamic nature i.e. many things happen at a certain time but at different places concurrently. This data is extensively huge to manage.  . | B | 6 |
| Real-world data needs more dynamic simulation and modeling, and for achieving the same, parallel computing is the key.  Parallel computing provides concurrency and saves time and money.  Complex, large datasets, and their management can be organized only and only using parallel computing’s approach.  Ensures the effective utilization of the resources. The hardware is guaranteed to be used effectively whereas in serial computation only some part of the hardware was used and the rest rendered idle.  Also, it is impractical to implement real-time systems using serial computing.multiple, smaller calculations broken down from an overall larger, complex problem | A | 8 |
| Applicaions | A+ | 10 |

**PART-B**

**Answer any four questions. Each question carries 3 weightage.**

|  |  |  |  |
| --- | --- | --- | --- |
| Question No | Marks/Grades allotment | Marks/Grades |  |
| 8  Ans | Recite PRAM model in details.  Parallel Random Access Machine, also called PRAM is a model considered for most of the parallel algorithms. It helps to write a precursor parallel algorithm without any architecture constraints and also allows parallel-algorithm designers to treat processing power as unlimited. It ignores the complexity of inter-process communication. | A | 12 |
| Figure: | A+ | 15 |
| 9  Ans | Illustrate dynamic programming |  |  |
| Dynamic programming is a technique that breaks the problems into sub-problems, and saves the result for future purposes so that we do not need to compute the result again. The subproblems are optimized to optimize the overall solution is known as optimal substructure property. The main use of dynamic programming is to solve optimization problems. Here, optimization problems mean that when we are trying to find out the minimum or the maximum solution of a problem. The dynamic programming guarantees to find the optimal solution of a problem if the solution exists. | A | 12 |
| Example: Fibonacci series | A + | 15 |
| 10.  Ans. | Demonstrate the backtracking designing techniques  Backtracking Algorithm tries each possibility until they find the right one. It is a depth-first search of the set of possible solution. During the search, if an alternative doesn't work, then backtrack to the choice point, the place which presented different alternatives, and tries the next alternative. | A | 12 |
| Example: | A + | 15 |
| 11.  Ans | Assess masters theorem. Find the complexity of recurrence relation. |  |  |
| If a ≥ 1 and b > 1 are constants and f(n) is an asymptotically positive function, then the time complexity of a recursive relation is given by  T(n) = aT(n/b) + f(n)  where, T(n) has the following asymptotic bounds:  1. If f(n) = O(nlogb a-ϵ), then T(n) = Θ(nlogb a).  2. If f(n) = Θ(nlogb a), then T(n) = Θ(nlogb a \* log n).  3. If f(n) = Ω(nlogb a+ϵ), then T(n) = Θ(f(n)).  ϵ > 0 is a constant. | A | 12 |
| Example | A+ | 15 |
| 12.  Ans. | Critique Strassen’s matrix multiplication algorithm with an example.  divide and conquer method, the main component for high time complexity is 8 recursive calls. The idea of Strassen’s method is to reduce the number of recursive calls to 7. O(nlog2 7).Strassen’s method is similar to above simple divide and conquer method in the sense that this method also divide matrices to sub-matrices of size N/2 x N/2 as shown in the above diagram, but in Strassen’s method, the four sub-matrices of result are calculated using following formulae. | B | 9 |
| M1 = (A11 + A22) (B11 + B22)  M2 = (A21 + A22)B11  M3 = A11(B12 − B22)  M4 = A22(B21 − B11)  M5 = (A11 + A12)B22  M6 = (A21 − A11)(B11 + B12)  M7 = (A12 − A22)(B21 + B22)  C11 = M1 + M4 − M5 + M7  C12 = M3 + M5  C21 = M2 + M4  C22 = M1 − M2 + M3 + M6  W(n) = 7W(n/2) + O(n2) =⇒ W(n) = O(nlog2 7). | A | 12 |
| Example | A+ | 15 |
| 13  Ans. | Describe the difference between NP complete and NP hard problems  A problem is in the class NPC if it is in NP and is as hard as any problem in NP. A problem is NP-hard if all problems in NP are polynomial time reducible to it, even though it may not be in NP itself.    If a polynomial time algorithm exists for any of these problems, all problems in NP would be polynomial time solvable. These problems are called NP-complete. The phenomenon of NP-completeness is important for both theoretical and practical reasons. | A | 12 |
| Diagram+ Examples of NP Complete and NP Hard problems | A+ | 15 |
| 14.  Ans | Analyse speed up,scalability,amdhals law.  Speedup- Speedup is defined as the ratio of performance for the entire task using the enhancement and performance for the entire task without using the enhancement or speedup can be defined as the ratio of execution time for the entire task without using the enhancement and execution time for the entire task using the enhancement. If Pe is the performance for the entire task using the enhancement when possible, Pw is the performance for the entire task without using the enhancement, Ew is the execution time for the entire task without using the enhancement and Ee is the execution time for the entire task using the enhancement when possible then, Speedup = Pe/Pw or Speedup = Ew/Ee Amdahl’s law uses two factors to find speedup from some enhancement:  Amdahl’s law is a principle that states that the maximum potential improvement to the performance of a system is limited by the portion of the system that cannot be improved. In other words, the performance improvement of a system as a whole is limited by its bottlenecks.  The law is often used to predict the potential performance improvement of a system when adding more processors or improving the speed of individual processors. It is named after Gene Amdahl, who first proposed it in 1967.  The formula for Amdahl’s law is:  S = 1 / (1 – P + (P / N))  Where:  S is the speedup of the system  P is the proportion of the system that can be improved  N is the number of processors in the system  Scalability or scaling is widely used to indicate the ability of hardware and software to deliver greater computational power when the amount of resources is increased | A+ | 15 |
| Strategy evaluation includes three basic activities: (1) examining the underlying bases of a firm's strategy, (2) comparing expected results with actual results, and (3) taking corrective actions to ensure that performance conforms to plans. |  |  |

**PART-C**

**Answer any two questions. Each question carries 5 weightage.**

|  |  |  |  |
| --- | --- | --- | --- |
| Question No | Marks/Grades allotment | Marks/Grades |  |
| 15.  Ans. | List out different problems types. |  |  |
| Any 3 problems +explanation | B | 15 |
| Any 5 problems+ explanation | A | 20 |
| Sorting: The sorting problem is to rearrange the items of a given list in nondecreasing order. sorting makes many questions about the list easier to answer. The most important of them is searching:  Searching: The searching problem deals with finding a given value, called a search key, in a given set (or a multiset, which permits several elements to have the same value).  String processing: A string is a sequence of characters from an alphabet. Strings of particular interest are text strings, which comprise letters, numbers, and special characters; bit strings, which comprise zeros and ones; and gene sequences, which can be modeled by strings of characters from the four-character alphabet {A, C, G, T}  Graph problems: One of the oldest and most interesting areas in algorithmics is graph algorithms. Informally, a graph can be thought of as a collection of points called vertices, some of which are connected by line segments called edges.  Combinatorial problems: From a more abstract perspective, the traveling salesman problem and the graph-coloring problem are examples of combinatorial problems.  Geometric problems: Geometric algorithms deal with geometric objects such as points, lines, and poly-gons.  Numerical problems: Numerical problems, another large special area of applications, are problems that involve mathematical objects of continuous nature: solving equations and systems of equations, computing definite integrals, evaluating functions, and so on. | A+ | 25 |
| 16.  Ans. | Demonstrate different algorithm design techniques |  |  |
| Any 2 techniques +demonstration with problem | C | 10 |
| Any 3 techniques +demonstration with problem. | B | 15 |
| Any 4 techniques +demonstration with problem | A | 20 |
| Brute force method, divide and conquer, greedy method, dynamic programming, back tracking, branch and bound  Explain technique with the help of a suitable problem | A+ | 25 |
| 17.  Ans | Assess different ratio theorm |  |  |
| Big -oh, Big Omega and Big theta theorems | A | 20 |
|  | Big -oh, Big Omega and Big theta theorems, mathematical notations, graph explanation | A+ | 25 |
| 18.  Ans | Criticize different methods to solve recurrence equation. |  |  |
| Any one method | C | 10 |
| Any two methods | B | 15 |
|  | Any three methods |  |  |
|  | There are four methods for solving Recurrence:  Substitution Method   * Equation: * Example:   Iteration Method   * Equation: * Example:   Recursion Tree Method   * Equation: * Example:   Master Method   * Equation: * Example: | A+ | 25 |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***