|  |  |
| --- | --- |
|  | **Pandit Deendayal Energy University**  **School of Technology**  **Department of Computer Science & Engineering**  Odd Semester 2022-2023 |

List of Practical for DAA

**Practical No. 1 and 2**

Implement the following sorting in any programming language.

1. Insertion sort
2. Selection sort
3. Merge sort
4. Quick sort

Now, measure the execution time and the number of steps required to execute each algorithm in best case, worst case, and average case.

**Practical No. 3**

Use singly linked lists to implement integers of unlimited size. Each node of the list should store one digit of the integer. You should implement addition, subtraction, multiplication, and exponentiation operations. Limit exponents to be positive integers.

What is the asymptotic running time for each of your operations, expressed in terms of the number of digits for the two operands of each function?

**Practical No. 4**

Implement a city database using unordered lists. Each database record contains the name of the city(a string of arbitrary length) and the coordinates of the city expressed as integer x and y coordinates.Your program should allow following functionalities:

1. Insert a record,
2. Delete a record by name or coordinate,
3. Search a record by name or coordinate.
4. Pint all records within a given distance of a specified point.

Implement the database using an array-based list implementation, and then a linked list implementation. Perform following analysis:

1. Collect running time statistics for each operation in both implementations.
2. What are your conclusions about the relative advantages and disadvantages of the two implementations?
3. Would storing records on the list in alphabetical order by city name speed any of the operations?
4. Would keeping the list in alphabetical order slow any of the operations?

**Experiment No. 5 [Greedy Approach]**

Implement interval scheduling algorithm.Given events with their starting and ending times, find a schedule that includes as many events as possible. It is not possible to select an event partially. For example, consider the following example:

|  |  |  |
| --- | --- | --- |
| **Event** | **Starting time** | **Ending time** |
| A | 1 | 3 |
| B | 2 | 5 |
| C | 3 | 9 |
| D | 6 | 8 |

Here, maximum number of events that can be scheduled is 2. We can schedule B and D together.

**Practical No. 6 [Divide and Conquer]**

Implement both a standard matrix multiplication algorithm and Strassen’s matrixmultiplication algorithm. Using empirical testing, try and estimate the constant factors for the runtimeequations of the two algorithms. How big must be before Strassen’s algorithm becomes more efficientthan the standard algorithm?

**Experiment No. 7 [Dynamic Programming]**

Implement the Floyd Warshall Algorithm for All Pair Shortest Path Problem.You are given a weighted diagraph , with arbitrary edge weights or costs between any node and node .Find the cheapest path from every node to every other node. Edges may have negative weights.Consider the following test case to check your algorithm:

|  |  |  |
| --- | --- | --- |
| ***v*** | ***w*** |  |
| 0 | 1 | -1 |
| 0 | 2 | 4 |
| 1 | 2 | 3 |
| 1 | 3 | 2 |
| 1 | 4 | 2 |
| 3 | 2 | 5 |
| 3 | 1 | 1 |
| 4 | 3 | -3 |

**Practical No. 8 [Backtracking]**

Solve the queens’ problem using backtracking. Here, the task is to place chess queens on an x board so that no two queens attack each other. For example, following is a solution for the 4 Queen’ problem.

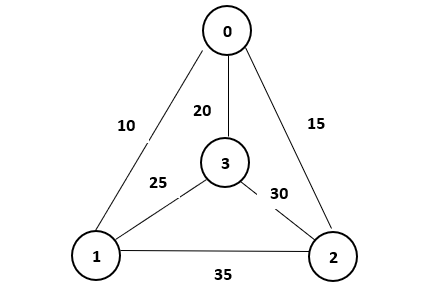
|  |  |  |  |
| --- | --- | --- | --- |
|  | Q |  |  |
|  |  |  | Q |
| Q |  |  |  |
|  |  | Q |  |

**Practical No. 9 [Branch and Bound]**

Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible tour that visits every city exactly once and returns to the starting point.

Solve this problem using branch and bound technique.

For example, consider the following graph:



A Travelling Salesman Problem (TSP) tour in the graph is . The cost of the tour is .

**Practical No. 10**

To design and solve given problems using different algorithmic approaches and analyze their complexity.

1. Your friends are starting a security company that needs to obtain licenses for different pieces of cryptographic software. Due to regulations, they can onlyobtain these licenses at the rate of at most one per month.Each license is currently selling for a price of $100. However, they are all becoming more expensive according to exponential growth curves: inparticular, the cost of license increases by a factor of each month, where is a given parameter. This means that if license is purchased months from now, it will cost. We will assume that all the price growth rates aredistinct; that is, for licenses (even though they start at the sameprice of $100).

The question is: Given that the company can only buy at most one licensea month, in which order should it buy the licenses so that the total amount ofmoney it spends is as small as possible?

Give an algorithm that takes the rates of price growth , andcomputes an order in which to buy the licenses so that the total amount ofmoney spent is minimized. The running time of your algorithm should bepolynomial in .

1. Suppose you are given an array with entries, with each entry holding adistinct number. You are told that the sequence of values is unimodal.That is, for some index between and , the values in the array entries increase up to position in and then decrease the remainder of the way until position . (So if you were to draw a plot with the array position on the-axis and the value of the entry on the -axis, the plotted points wouldrise until -value , where they’d achieve their maximum value, and then fall fromthere on). You’d like to find the “peak entry” without having to read the entirearray - in fact, by reading as few entries of as possible. Show how to find the entry by reading at most entries of .