DESIGN AND ANALYSIS OF ALGORITHMS

Project Report

Data Storage:

-This was done through file reading and during the file reading process, 2 different adjacency lists were created.

-The first list has all the countries at the head of the lists and all their related data stored in the link list after it.

-The second list has all the dates at the head of the lists and all the countries with their cases on those dates so all queries related to dates could be done in lesser amount of time.

Algorithms used in queries:

Query1: (Top 20 countries with the most confirmed cases on a given date)

-First the algorithm checks whether the date entered by the user exists or not and if it doesn't it displays an error message.

-If the date exists then the Date List is used to traverse through all dates till it finds the one specified by the user.

-Upon finding the date entered by the user, bubble sort algorithm is used to sort the countries in decreasing order according to their confirmed cases.

-After sorting, first 20 countries will be at the start of that particular Date List index and they are displayed one by one.

-This will take time complexity O(n*m) as traversing through the adjacency list will not cost less than this.

Query2: (Country with the highest new cases between 2 given dates)

-First the algorithm will check whether the dates entered by the user exist or not and if they don't then an error message will be displayed.

-If the dates exist then a vector of type pair is made and the pair is of type string and integer (vector<pair<string,int>>). The size of the vector will be the amount of countries present and the vector of type pair will hold a country and its new cases added up.

-This will be done by traversing the Country List and checking if the date lies between the user given dates then it will add its confirmed cases till each country has been traversed.

-Then the vector will be sorted in decreasing order and the country(s) with the highest new cases that fall within the given dates will be outputted.

-This will also take O(n*m) time complexity as traversing through the adjacency list cannot cost less than this.

Query3: (Longest spread period of a country)

-This problem has been solved using dynamic programming by the longest increasing subsequence method.

-The program checks if the country exists or not.

-A vector is used and it keeps storing all the cases day by the in an increasing order.

-Alongside this vector, another temporary vector is used for the sake of comparing the length of the sequence so that the vector with the longer subsequence is saved.

-In this way the start date, end date and as well as the span (number of days) is stored and outputted.

-This has been solved by dynamic programming method and takes up complexity of O(nlogn)

Query4: (Longest decreasing death toll)

-This problem has been solved using dynamic programming by the longest decreasing subsequence method.

-The program checks if the country exists or not.

-The print_all() function has been created which basically used a lot of vectors and keeps updating them in the order of decreasing deaths.

-The final vector will be the most updated vector which will have the highest length containing the longest death toll for that specific country.

-The number of days will be the length of the vector and the content of the vector will be the amount of deaths in decreasing order.

- This has been solved by dynamic programming method and takes up complexity of O(n^2).

Query5: (Countries selected for aid)

-This problem has been solved using dynamic programming by the knapsack problem method.

-The whole functionality of the knapsack problem has been implemented in this query.

-A matrix is created and is updated step by step until every value of the matrix has been updated.

-The last row and last column will contain the final score.

-Then using traceback method we will find which countries have been selected for aid as well as displaying their costs.

- This has been solved by dynamic programming method and takes up complexity of O(nlogn).

Query6: (Similar active cases between 2 countries)

-This problem has been solved using dynamic programming by the longest common subsequence method.

-Firstly, the program will check whether both countries exist or not.

-The Country List will be traversed and 2 vectors are used which will hold the active cases of the countries on the same days.

-Once the vectors have stored all the cases on the similar dates, a matrix is created and it contains amount of active cases on different days. The highest value in the matrix will indicate the highest possible common subsequence between 2 countries.

-From that using traceback method we will find the dates on which their cases were similar and then display them to the user.

This has been solved by dynamic programming method and takes up complexity of O(n*m).