DESIGN AND ANALYSIS OF ALGORITHM PROJECT REPORT

## Members : 1. Fahad Tahir (18K-0286)

## 2. Risha khan (18K-1354)

**ABSTRACT:**

Throughout our course we had discussed many problems that we encounter in our daily life such as knapsack, integer partition, rod cutting problem and many more. The main aim of this project was to implement these algorithms and show an interactive way of solving these problems efficiently and quickly. The problems are divided in 3 groups according to the problem types they solve or belong to.

1. Mathematics
2. Sets
3. Strings

The mathematics section deals with problems such as chain matrix multiplication, coin change making, integer partition, rod cutting problem. After giving it the sample it will give u a simple output of number of possibilities.

The sets section deals with problems such as 0/1kapack problem, longest increased sub subsequence, longest common subsequence, shortest common subsequence. It gives a simple output along with some graphical representation of the values that will change according to the altering of the values being provided.

The string section solves problems such as levenshtein distance and word break problem. This will give you a simple output of the distance or where the word breaks.

Furthermore, there are also 10 samples given to test how the algorithm works and you can also input your own values.

**INTRODUCTION:**

This section gives the introduction of all the problems that rare covered I his project and a short detail of what these problems actually are.

1. Mathematics :

# Longest Common Sequence:

These are the classic computer science problems having two sequences from which we are to find the one with the longest length subsequence. It is the basis of data comparison programs. Here the subsequence is a particular order appearing in the same relative order but isn’t necessarily adjacent.

# Longest Increasing Subsequence:

The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given order in a way that all elements of the subsequence are sorted in the ascending order. It can be both adjacent or unique, it varies at times.

# Shortest Common Super sequence:

In Shortest Common Super sequence, we are given two strings str1 and str2, the task is to find the length of the shortest string that has both str1 and str2 as sub sequences.

# Balanced Partition:

Given a set of integers, partition those integers into two parts where the difference between the two parts is minimum. This problem is known as balanced partition problem.

1. Sets :

# Chain Matrices Multiplication:

These are sequences of Matrices and the main aim is to find the most well organized and fastest way to solve these matrices by multiplying them. Moreover, deals with finding the best way of multiplying a set of matrices.

# Coin Change Making:

The Coin Change Problem is considered by many to be essential to understanding the paradigm of programming known as Dynamic Programming. The two often are always paired together because the coin change problem encompass the concepts of dynamic programming.

# Integer Partition:

In number theory and combinatorics, a partition of a positive integer n, also called an integer partition, is a way of writing n as a sum of positive integers. Two sums that differ only in the order of their summands are considered the same partition

# Rod Cutting:

Given a rod of length n inches and an array of prices that contains prices of all pieces of size smaller than n. Determine the maximum value obtainable by cutting up the rod and selling the pieces.

1. Strings :

# Levenshtein Distance:

The Levenshtein distance is a string metric for measuring difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e., insertions, deletions or substitutions) required to change one word into the other.

Word Break:

Given an input string and a dictionary of words, find out if the input string can be segmented into a space-separated sequence of dictionary words.

**PROPOSED SYSTEM**

We made a website based on all the algorithms their solutions with examples to make it easier to explore them all, at one place with some examples, graphs, charts and needed representations.

**EXPERIMENTAL SETUP**

SAMPLES:

a) Longest Common Subsequence,

[["AFAFAHHAHFADAFAFTAHHAHHAFAFFDAFAFHADDAAFAFFAFADAFAFAHHAHHADDAAFAFFAHDAADFARDHDAIADAAD", "FATDAFAFAHHAHHADDAAFAFFAHDAADFADAFAFAHHAHFADAFAFAHHAHHADDAAFAFFDAFARRRRFHADDAAFAFFAHDAADHADAAID", ],

...

]

b) Shortest Common Super Sequence( same as a)

c) Levenshtein Distance (edit-distance) ( same as a)

d) Longest Increasing Subsequence

[[ 10, -68, 7, 80, 82, 90, 8, -39, -36, 73, 12, 92, 100, -64, -33, 86, 85, 1, -34, 91, 74, -37, 42, 57, -31, 55, 14, 82, 72, 46, 70, 10, 76, -67, -62, 28, 8, 86, -32, -65, 48, -60, 23, 2, -35, -67, 96, 5, 89, 12, -36, 53, 19, 94, 76, 72, -39, -69, 29, 27, 92, 89, 58, 22, 94, -38, -66, 95, 46, 27, 85, 82, -32, 14, -32, 50, 51, 9, 43, -34, 4, -65, 86, 94, 28, 29, -35, 50, 74],

…];

e) Chain Matrix Order (same as d just no negative values)

f) 0-1-knapsack-problem

[for set of points and weights same from e shuffled up , W = 286

g) Partition-problem (same as d)

h) Samples for Coin-change-making-problem

[

[ [ same as f ], 354 // roll number]

…

];

i) Sample for Word Break Problem

[[‘sdflis algier gosls alsidf glslit slfiigne glsfaig slifige gliseingns sdigse’, ‘fahadtahir’ ]]

**RESULT AND DISCUSSIONS:**

Based on the calculations, graphs, outcomes and results we concluded that the results were 100% correct.

**CONCLUSION**

The outcome of the project came out to be a complete solution for dealing with complicated algorithms with examples, charts, needed representations, analysis, theory, and variations of all major famous and daily life used problem’s solution