Project Report

Design and Analysis of Algorithms CS302

Group Member(s): Syed Abdullah Muzaffar (18K-0169)

Section: 5H

Instructor: Muhammad Sohail Afzal

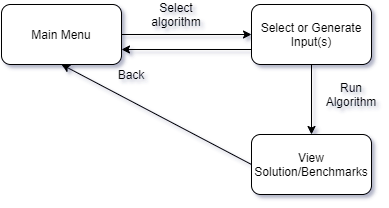
**Abstract**:

An implementation of some of the dynamic programming algorithms that were taught in the course.

**Introduction:**

An app that provides users the ability to run these algorithms on their smartphones, and either select from a preset 10 input cases, or generate some randomly.

**Proposed System:**



The target platform is android, with flutter as the framework.

The user starts off at the main menu, where they can select an algorithm, and run the valid input cases (or generate random ones). The selected Input cases, solution and Time taken are presented. The user can return to the main menu to select another algorithm anytime.

The UI that is presented to the user is extremely minimalistic and simple, with a small color palette, to avoid thematic clashes, and follows the material guidelines.

**Experimental Setup:**  
The Inputs are divided into 4 Categories, which we shall label (A,B,C,D). All Inputs allow for random generation.

**Category A:**

Applicable for:

* "Longest Common Subsequence"
* "Shortest Common Supersequence"
* "Levenshtein(Edit) Distance"

The Input Dataset consists of 10 Text Files(1-10.csv), which contain two string sequences using alphabets of the name in random order and frequency, of length a random number between 30 to 100 characters.

e.g. name = ABDULLAH

String X = HBLBDADBADABABDLLUDDUHALHDLAHALLABDHUHLHLHAH

String Y = LLUBDAHUHLLABDALLBUDUAHHLBDDBAHAAALAUU

String X = BADLLALLADLLAALUALLULDULBHALHUABHLDDHLA

String Y = LLLALBDDADLALLBDDUBUDBABAHHHHLHLLHLULAHHBBLUD

**Category B:**

Applicable for:

* "Longest Increasing Subsequence"
* "Matrix Chain Multiplication"
* "Partition Problem"
* "Coin Change Problem"

A sequence of n random numbers from 0 to 100, with n a random number between 30 to 100 integers.

e.g.

Sequence = 72, 22, 25, 15, 64, 83, 84, 16, 94, 24, 71, 84, 42, 82, 76, 60, 84, 64, 67, 39, 58, 24, 18, 92, 17, 79, 71, 13, 67, 47, 30, 42, 73, 31, 79, 36, 33, 49

Sequence = 69, 20, 25, 76, 91, 81, 30, 53, 67, 34, 85, 80, 12, 91, 69, 30, 17, 92, 42, 37, 84, 56, 33, 81, 15, 62, 39, 94, 16, 55, 92, 25, 25, 22, 26, 25, 40, 11

**Category C:**

Applicable for:

* "0-1 Knapsack"
* "Rod Cutting"

Two sets of n points (n is a random number varying from 10 to

100), with values ranging from 1 to 100.

The capital W is the last three digits of the roll number.

Roll No. 18K-0169 -> W = 169.

e.g.

1a= [85, 55, 17, 47, 70, 70, 71, 36, 63, 51, 39, 55, 82, 44, 48, 80]

1b= [96, 19, 14, 18, 53, 89, 97, 63, 55, 51, 29, 61, 22, 4, 91, 45]

2a= [26, 98, 9, 88, 71, 33, 7, 11, 99, 38, 75, 1]

2b= [54, 30, 89, 46, 53, 14, 20, 10, 15, 28, 70, 38]

**Category D:**

Applicable for:

* "Word Break"

A set of randomly generated strings from alphabets a-z, and the input as full name in lowercase.

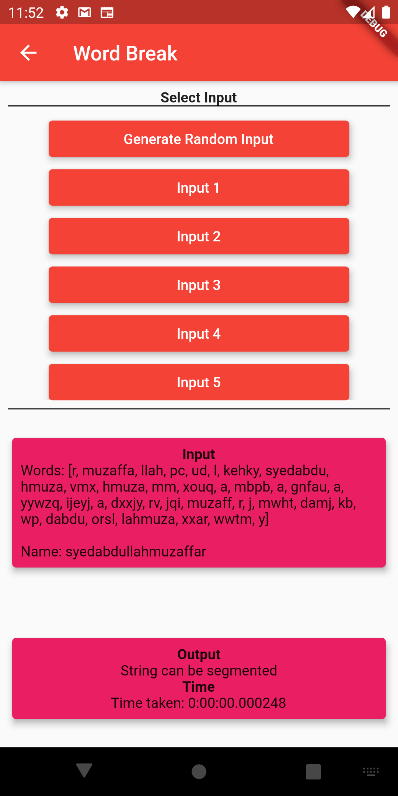
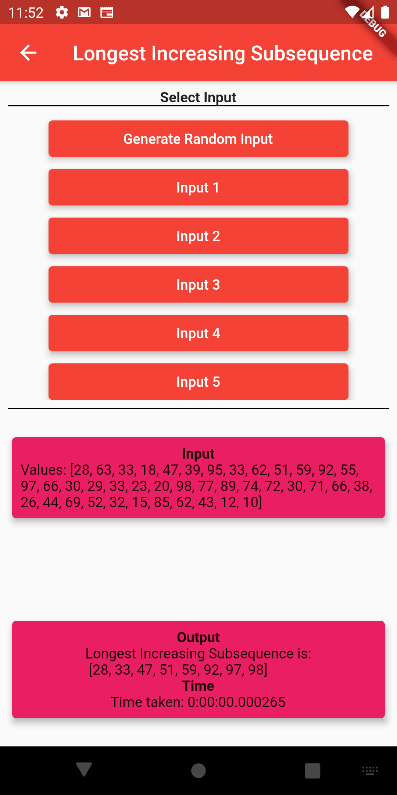
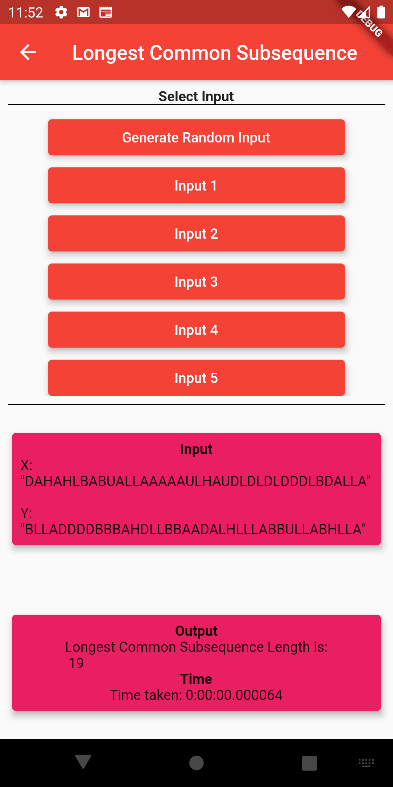
e.g.

name = syedabdullahmuzaffar

Dict = ["lvw","m","ahmuza","vcgg","amwj","j","uza","uzaff","dabdu","b","dlzk","dabdull","k","h","r","p","uzaf","ol","l","bdu","rpe","a","xfwav","ubevy","wxs","hmuzaf","za","hfr","f","nm","iik"]

Dict = ["gg","yedabdu","i","affa","ffa","oqxj","d","laq","llahmu","oujcs","kjwwy","l","yedab","uza","isug","edabdu","ja","llahmuz","syedabdu","a","uf","a","e","nhbwh","mldwp","hxrl","ff","syeda","ookox","edabdul","eda","lahm","ilie"]

**Results and Discussion:**



The program provides valid outputs on valid inputs, and the generator works as expected(3rd screen). The Dynamic nature of the algorithms provide a solution in optimal time(in the order of milliseconds).

**Conclusion:**

This Project proves that algorithms are mostly platform-agnostic(not hardware), and that these dynamic algorithms may provide a speedup on mobile applications.

**References:**

1. <https://medium.com/techie-delight/top-10-dynamic-programming-problems-5da486eeb360>
2. https://flutter.dev/docs/development/ui/widgets/material