**DAA PROJECT**

THE PROJECT REPORT

***Submitted by***

BL.EN. U4AIE19028 K.Vishnu Sainadh

BL.EN. U4AIE19034 K.Satwik

BL.EN. U4AIE19066 V.Ashrith

**For the course**

# 19AIE212- Design and Analysis of Algorithms

**Guided and Evaluated by**

***Dr. Supriya M***

***Asst. Prof (SG),***

***Dept. of CSE***,



AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

May-2021

**AMRITA VISHWA VIDYAPEETHAM**

Title of the project

**DAA PROJECT**

**Acknowledgement**

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people whom made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranamas at the lotus feet of “AMMA”, MATA AMRITANANDAMAYI DEVI who showered her blessing upon us throughout the course of this project work.

It is a great pleasure to express our gratitude and indebtedness to our project guide Dr.Supriya M , Assistant Professor, Department of Computer Science and Engineering, Amrita School of Engineering, Bangalore for your valuable guidance, encouragement, moral support and affection throughout the semester.

#### TABLE OF CONTENTS:

1). PROBLEM STATEMENT………………………….5

2). IMPLEMENTATION ………………………………5

4). GUI………………………………………………….7

3). RESULT ……………………………………………8

**TABLE OF FIGURES:**

1. Fig 4.1 : GUI with input in the lineedit fields…………8
2. Fig 4.2 : After giving inputs all the inputs will be displayed …………….9
3. Fig 4.3 : Result will be displayed in a new dialog box ………….9

**Problem statement:**

A firm has divided its marketing area into n zones. The amount of sales depends upon the number of the salesman in each zone. The firm has been collecting the data regarding sales and salesman in each area over a number of past years. For the next year firm has only x number of salesman and the problem is to allocate these salesman to n different zones so that the total sales are maximum.

**Implementation:**

For this problem we are going to use dynamic programming approach

Dynamic programming is a very powerful, general tool for solving optimization problems.We can use Dynamic Programming to optimise any recursive solution that has repeated calls for the same inputs. The idea is to actually save the effects of sub problems so that we don't have to recalculate them later. The time complexity of this simple optimization is reduced from exponential to polynomial.

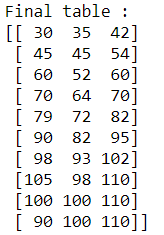
Procedure adopted in Dynamic Programming:

* Define the variables, objective function and constraints.
* Divide the problem into number of sub-problems.
* Develop recursive relationship for optimality.
* Decide whether to follow the forward or the backward method to solve the problem.
* Make tabular presentation to show the required values and calculations for each stage.
* Find optimal policy at each stage and then the overall optimal policy.

Let us take a look at an example:

Let us take the number of zones to be three (3) and the number of salesman to be nine (9)

And the profit values to be as follows



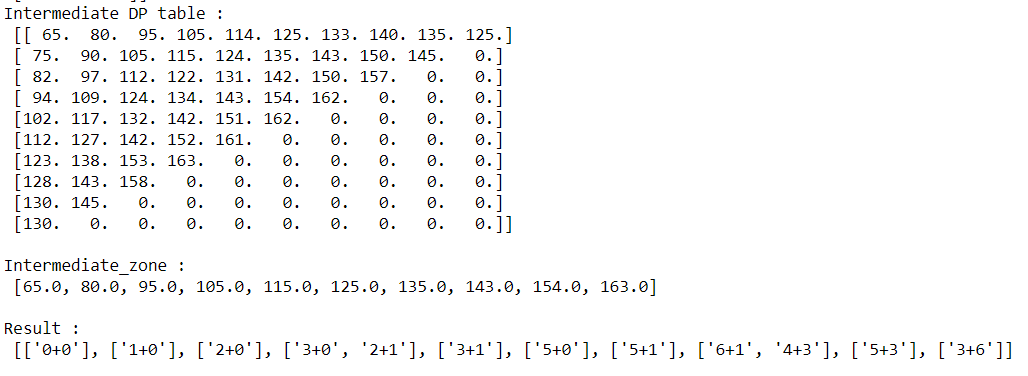
The table says that if there are zero salesman then first zone will get a profit of 30 , second zone will get a profit of 35 and third zone will get a profit of 42. Likewise for 1 to 9 salesmans the profit values will be as shown in table where each column represents each zone.

For this case we will divide the process into three steps

1. Stage 1: Zone1
2. Stage 2: Zone1+Zone2
3. Stage 3: Zone1+Zone2+Zone3

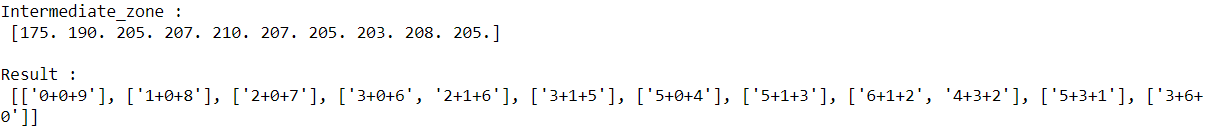
Here Stage 1 is nothing but the first column values in the profit values table.

In Stage 2 we need to write a DP table for Zone1 and Zone2 and the DP table will be as follows:



In the DP table, the total profit is calculated for each and every combination of number of salesman. Since the total number of salesman cannot exceed by 9 in a combination we have filled profit values with zeroes when it exceeds. For each combination the number of profit values will be increasing so we have to choose the maximum values from the diagonals. If there are two maximum values repeated both should be considered as they will have different combination. Those values are shown in the Intermediate zone array and the corresponding combinations are shown in Result array.

Coming to Stage 3 we are going to use the results of Stage 2



Here we got the Intermediate zone using Stage 2 results , as you can see in Stage 2 results for zero salesman the profit is 65 so nine (9) more salesman should to be added to make the total salesman as nine (9) so profit for nine salesman in zone 3 that is110 should to be added to total profit of first combination since we have added 9 salesman from zone 3 now the combination becomes [0,0,9]. Likewise all the profit values corresponding to the number of salesman will be added to the total profit to each combination such that the total sum of combination will be total number of salesman in this case it is nine (9).

And then we will take the maximum profit value and the combination from the results :



If number of zones is greater than or equal to three, First if it is three, DP table will be created for zone1 and zone2 and max values of profit will be taken and then third zone profit will be added in reverse to the profits in order to make the total sum of combination equal to number of salesman.

If number of zones is equal to four, DP table will be created for zone1 and zone2 and then DP table will be created for zone1+zone2 and zone3 and then at last fourth zone profit will be added in reverse to the profits in order to make the total sum of combination equal to number of salesman.

Coming to the complexity, as we are using Dynamic programming approach it will generally be polynomial. So, In our algorithm also the time complexity is polynomial.

**GUI:**

We have used pyqt5 module for our GUI. PyQt5 is one of the most common Python modules for creating graphical user interfaces, and this is due to its ease of use, as you can see. The PyQt5 creator, which makes it so easy to create complex GUI apps in a short amount of time, is another great feature that enables developers to use PyQt5. To build your shape, simply drag and drop your widgets.

PyQT is a toolkit for creating graphical user interfaces. It's a Python interface for Qt, a cross-platform GUI library that's one of the most relevant and widely used. PyQt5 is a Python module that helps us to quickly create graphical user interfaces. The key explanation is the speed at which you can create graphical user interfaces. It includes a fantastic tool called Qt Designer, which is a drag-and-drop interface that creates code for you based on the GUI you have. This makes it much easier to create attractive applications.

**RESULT:**

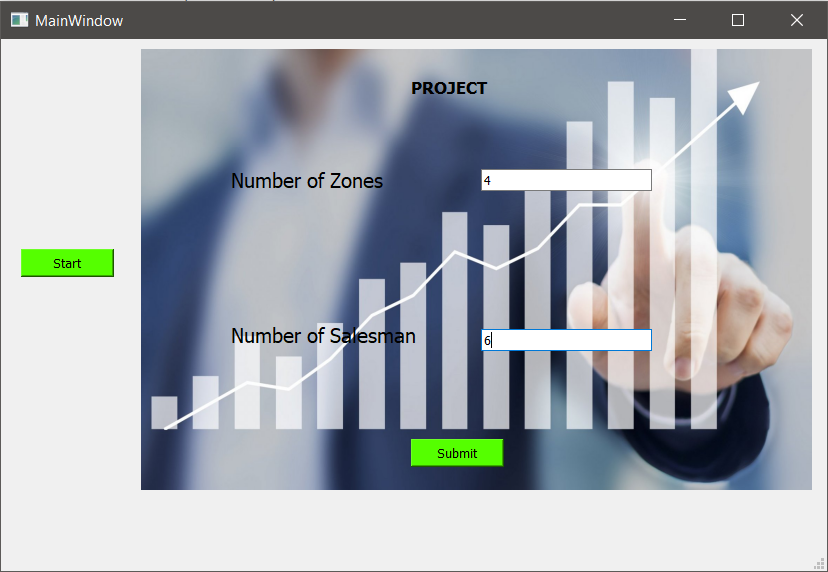


Fig 4.1 : GUI with input in the lineedit fields

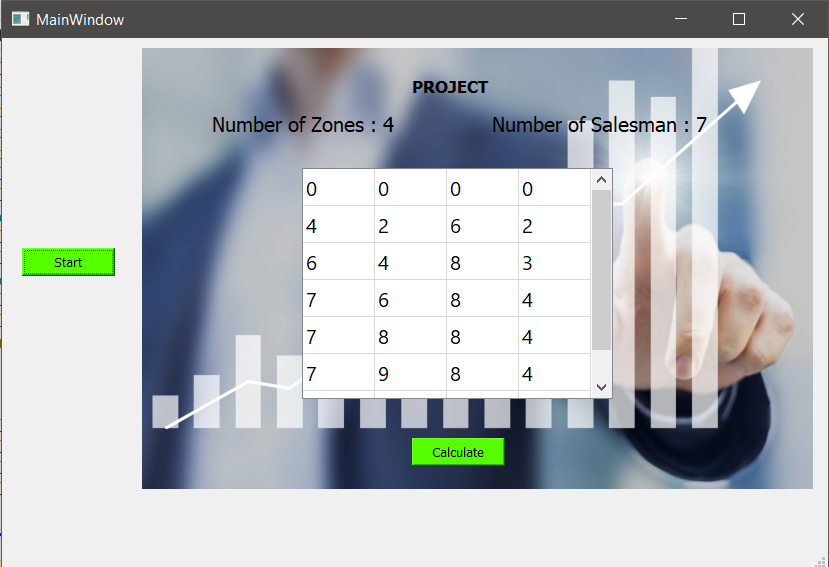


Fig 4.2 : After giving inputs all the inputs will be displayed

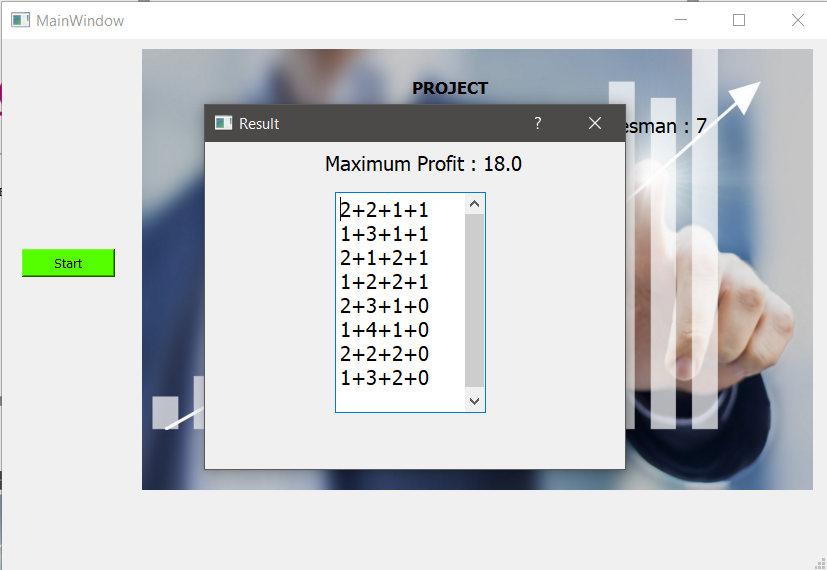


Fig 4.3 : Result will be displayed in a new dialog box