**Design and Analysis of Algorithms**

**Final project report**

**Project 1 – Course Matching System**

Team members:

Mustafa Muswadi

Ahmed Albishri

Khalid Almalki

Mohammed Almansoori

**Project 1 – Course Matching System**

**Assumptions:**

We have followed the technical requirements that have mentioned in the project description.

**Illustrating diagram:**

We have implemented an algorithm that find the best allocations for all professors and courses by using (Greedy Algorithm and Dynamic programming). The algorithm will find the best professors for courses based on the course content and the expertise of the professor. So, we will come up with every professor who will teach one course. Also, how we can modify the algorithm if a professor can teach two courses. At the end we came up with the maximizing of the totality of our course matching.

**System Architecture:**

As we know that, we have two requirements. The first requirement is every professor must teach a course and every course must be taught by a professor. The second requirement is if a professor must teach two courses. The systemarchitecture works as following steps. First of all, the **Main class** will send the request to the **Controller** and the **Controller** will call the data from **SCV files** and prepare it. So the **Controller** will send it back to the main class. Then, the **main class** will send it to the output screen to show it out.

Data

(SCV Files)

Main Class

Data Controller

Output of Dara

**Efficiency of Algorithms:**

As our code contains three nested loops such as the pattern in the file of First\_Req.py, that are:

s: is the subjects number

p: is the professors number

t: is the topics number

O(n) = O(s x p x t)

For n in general:

O(n) = O(n x n x n) = O(n^3)

* **Dynamic programming algorithm:**

**Based on the proposed design for five courses and five professors as shown in the following table.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Course1 | Course2 | Course3 | Course4 | Course5 |
| Ahmed | Web 64 | Pro 53 | BD 47 | NA 46 | DS 44 |
| Khalid | Web 84 | NA 60 | Pro 41 | DS 40 | BD 28 |
| Mouhammed | Web 70 | NA 70 | DS 60 | Pro 56 | BD 32 |
| Mustafa | Web 72 | NA 58 | DS 44 | Pro 39 | BD 38 |
| Talha | DS 80 | Pro 71 | BD 39 | Web 36 | NA 33 |

We modeled the professors’ names in a dynamic programming table, so we can apply the dynamic programming approach. We are interested in each step and cell of this table to find the maximum value for the professors’ expertise. For instance, the cell Ahmed & Ahmed has 0 value as we are not interested to compare the same professor with himself, and the same case for all the main diagonal diameter. However, for the cell Ahmed and Khalid cell (Ahmed is the row and Khalid is the column), we are finding the value of the sum of the maximum courses for each of the involved professors, and here they are Ahmed and Khalid. We are considering to not taking the same courses in each case and considering the next maximum.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | Ahmed | Mustafa | Khalid | Mouhammed | Ali |
| Ahmed | 0 | (Pro, Wed)  53+72= 125 | Pro (DS, Wed)  53+142=195 | Pro (NA, Wed, DS)  53+212=265 | BD(NA, Pro, Web, DS)  47+283=330 |
| Mustafa |  | 0 | (DS, Wed)  58+84= 142 | NA (Web, DS)  56+154=212 | NA(Pro, web, DS)  58+225=283 |
| Khalid |  |  | 0 | (Web, DS)  84+70=154 | Pro (Web, DS)  71+154=225 |
| Mouhammed |  |  |  | 0 | (Web, Pro)  70+80= 150 |
| Talha |  |  |  |  | 0 |

**GitHub:**

<https://github.com/AhmedAlbishri/CS5592_Project>

**References:**

* Stack Overflow
  + - <https://stackoverflow.com>
* GeeksforGeeks:
  + Dynamic Programming | Set 8 (Matrix Chain Multiplication)
* <http://www.geeksforgeeks.org/dynamic-programming-set-8-matrix-chain-multiplication/>