# Assignment -2 PS4 (Student Dissertation)

# Algorithm Design Document

**Purpose**

The purpose of this document is to:

* Provides the details of the algorithm design
* Provide time complexity of the algorithm

As part of the problem statement we’re asked to formulate an efficient algorithm using dynamic programming which calculates the number of unique allocation of topics to the students such that everybody gets exactly one topic of their liking and no two students get the same topic allocated.

**Design details**

If we solve this problem using brute force the time complexity will be O(n!) which is very high, and therefore instead of using brute force we can solve this problem using dynamic programming as asked in the problem statement. Here, we’re using bit mask which is a binary number to represent the state of allocation.

Dynamic Programming is mainly an optimization over plain recursion. the idea is if we have a recursive solution that has repeated calls for the same input, we can optimize it using dynamic programming, that is to simply store the results of subproblems, so that we do not have to re-compute them when needed later, This simple optimization reduces the time complexity to O(n2n).

**Implementation details of methods in the class assign\_subject**

**Method 1:** **\_\_init\_\_()**

Initializes the variables all mask and total subjects

**Time Complexity:** As the method doesn’t have any iteration or recursion the time complexity for this method is constant, therefore the time complexity will be order of O(1).

**Method 2:** **count\_allocations()**

Recursively counts the total possible allocations and returns the result, the way it achieves this is using dynamic programming to store the result of subproblems so that we do not have to re-compute them when needed later.

**Time Complexity:** This method uses recursion, the time complexity is O(n2n).

**Method 3:** **get\_total\_allocations()**

Initializes the variable allmask and creates a list of lists allocations with initial value as -1 for each element.

**Time Complexity:** As the method contains iterations to initialize the allocations the time complexity is O(n2n).