# Design

This algorithm uses 3 methods,

**1. To read\_students\_pref**

Reads the input file and provides the list of students' preferences as array in the same student order as input.

**2. unique\_allocation**

It is a recursive method to calculate the total number of unique possiblity of assignment based on student preferences.

The input to this method is

preferencesList - List containing the student preferences for all student

StudentIndex - The index of the student from where we have to calculate the preferences (starting value would be 0)

electedTopics - List topics that are chosen by the previous student (starting value would be a empty list)

In this method,

1. if the student index reached the end, then will return the count as 1. (Which means a proper topic is chosen for all the students)

2. an appropriate topic will be chosen for the current student which is not chosen by previous students.

2.1. repeat the step 1 & 2

3. If no more possible topic is present in preferences, then return the so far computed possible path count.

**3. write\_to\_file**

Writes the provided data to file (To write the output to file).

# Complexity Analysis,

T(n) = T(n-1)\*Cn where n > 0

T(n) = 1 where n = 0

Cn - students preference count

n - number of students

Lets assume the maximum number of choices a student can have as n (for equal one on one allocation),

the BigO would be,

O(n) = n^n