**Part 3 (Choosing Team-Harshit Shiroiya)**

* **Set of Valid States**- There are a set of states, in the initial state there are no students assigned to any team. The input is taken where the students are asked for their group preferences and the list of members, they do not want to work with. The valid states are assigning the students in the groups in all possible ways i.e. group of one, two or three as per their preference.
* **Successor Function**- The successor function here is to assign the groups to all the students with consideration of their choices in the input.
* **Goal State**- The goal is achieved when all the students are assigned to team and such that the time spent by the AI’s is the least.
* **Cost Functions**-

1. Total Grading Time:
   * + The time taken to grade all the groups by the AI’s. It is 5 mins per group. So total will be Total no. of teams \* 5.
2. Different Group Size:

* The students who mentioned a group size in their preference and they get assigned a group size of not their choice. This will take 2 mins per student.

1. Members in Different Group but share their code.

* There is a probability of 5% that the students with different team share code which will lead to a cost of 60 mins. Therefore, 60\*0.05.

1. Student assigned a team member who they selected not to work with:

* If a student is assigned a team member who they opted not to work with then this will cost 10 mins.
* **Algorithm Working**
  + The search algorithm used here is Depth first Search as we firstly create random possibility of group formation. Later we, compute the cost for all the possibility and we yield the best cost that is the least cost calculated and display the same.
  + The student\_choice function first takes the given preferences of each student and returns the list of students along with their preferences in the form of dictionary.
  + Then we initiate create\_group function and possible\_team\_members wherein we create all permutations (one, two or three members) of the groups that can be formed in the preference form.
  + Then, The successor group creates a list where in it checks with all the previously assigned groups and looks if all the groups are full or not.
  + Then we compute the cost of all the permutation of assigned teams and yield the better solution.
* **Result**
  + We noticed that the first testcase which has six students had the minimum cost of 24 mins.
  + In the second testcase, we found the cost to be 43 mins and this test case had thirteen students
  + While in the third testcase there were twenty students and the cost turned out to be 66 mins which took a longer time to find this solution as this had many permutation of groups and students.