## Project 3 Report

We have solved the Constraint Satisfaction Problem of Course assignment to TAs using 3 methods which are listed below:

1. **Backtracking Search (BS):** In this method, naïve DFS is done to expand the nodes. At each level, it is checked whether all the constraints are satisfied or violated. If the constraints are satisfied and if either of courses or TAs are assigned completely, the algorithm exits. If the constraints are violated, the algorithm backtracks and tries another assignment. We used a scoring function, where in the TA with highest match of skills to requirements of class is assigned optimally.
2. **Forward Checking BS:** In this method, to find bad assignments much earlier, rather than expanding the tree to a larger depth and tracking back, certain checks are done, at each iteration. The check that domain of each of the TAs to be nonempty is made and if violated, backtracking is done. The code for this method is implemented in ***do\_fc()***.
3. **Arc Consistency + Forward Checking BS:** In this method, besides doing forward checking, Arc Consistency is also done. The approach follows a way wherein each of the neighbors of a TA are checked for a validity of domain. Initially all the neighbors are added to the queue and arc consistency is checked between each of the neighbors and the current TA. If there is a violation, the course is removed from domain of the TA and is added back to the queue. This process is done until we find all neighbors to be consistent or there is no way an assignment can be made consistent in which case it is backtracked. The code for Arc Consistency is preset in ***arc\_cons()*** and ***rm\_incons()***.

**Observations:**

With sample dataset, we observed that, 8 courses need not have TAs and 6 courses cannot be assigned with TAs due to lack of enough number of TAs.