**Assignment 5 Grading Scheme**

Implementation of Suffix tree 20 marks

Inefficient implementation (space Complexity O(n^2)) -5 marks

Without wildcards Patterns 20 marks

Question mark Patterns 20 marks

Inefficient (Question marks replaced by every possible character) -5 marks

Star Wild Card Patterns 20 marks

Viva 20 marks

Also, I have provided test cases and their solutions in the respective folders. These test cases contain different kinds of test cases.

1. corner\_1.txt , corner\_2.txt and very\_small.txt are very small test cases for checking the correctness. They may be manually looked into.
2. \*\_small.txt are test cases with lesser number of queries, in case student has written an inefficient solution.
3. Without\_wild\_cards.txt contain only text queries without any wildcard.
4. only\_star, only\_questionmarks are test cases if student has implemented only a single kind of wildcard.
5. mixed\_\*.txt contains every possible combination and are large files for checking the efficiency.

To test students output on the above test-cases, checker.py is provided.

$ python checker.py

The above command will compile students SuffixTree.java and output the student answers into a directory ‘student/’. Then it will compare model solutions in ‘solutions/’ directory and students answers and show pass/fail for each test case.

Please go through the code of checker.py to know exactly how it is working and tweak it as per your need.