

CSC 226 - SPRING 2017
ALGORITHMS AND DATA STRUCTURES II
PROGRAMMING ASSIGNMENT 4
UNIVERSITY OF VICTORIA

1 Programming Assignment

The town of NewVille has grown to be a big city now. But as with all the cities in the world, NewVille has many advantages to offer to its inhabitants at a high price. The crime rate has increased and the Police department is trying to update to cope up with the growing database of case records and criminal records. Your job is to implement an efficient algorithm that will help them to search for any string (may be a suspect's name, may be some incident, or a file number) in their database, so they can solve cases more quickly.

Input: A text file and a string or pattern to search in the text file.

Output: The index of the first occurrence of the pattern in the text if the text contains the pattern. Otherwise return the length of the text.

A Java template has been provided containing one empty function and an empty constructor. The constructor `KMP` takes a string representing the pattern we are searching for and constructs a DFA. The function `search` takes a string containing the whole text from the given text file, and returns the index of the first occurrence of the pattern in the text if the text contains the string/pattern, otherwise returns the length of the text.

You must use the provided Java template as the basis of your submission. You may not change the name (not even change uppercase to lowercase or vice versa), return type or parameters of those functions. The main function in the template contains code to help you test your implementation by reading it from a file. You can use the test cases uploaded on connex for Lab 9. You may modify the main function or any other function, because your submission will be tested using a different main function. You can use any helper methods or any helper classes. You can use any built-in class or write your own classes and data structures. We advise you to put all the classes you write in the same file, but no other class except the provided one should be declared as a public class.

2 Evaluation Criteria

The programming assignment will be marked out of 40, based on a combination of automated testing (using large texts) and human inspection.

You are advised to implement KMP algorithm. The running time of your code should be at most $O(M + N)$, where M is the length of the text you are searching and N is the length of the pattern you are searching for, for each pattern. The mark for each submission will be based on

both the asymptotic worst case running time and the ability of the algorithm to handle inputs of different sizes.

Score	Description
0 - 15	Submission does not compile or does not conform to the provided template or crashes for any of the data sets.
15 - 40	The implemented algorithm is $O(M + N)$ and gives the correct answer on all tested inputs.

To be properly tested, every submission must compile correctly as submitted, and must be based on the provided template. If your submission does not compile for any reason (even trivial mistakes like typos), or was not based on the template, it will receive at most 15 out of 40. The best way to make sure your submission is correct is to download it from `conneX` after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. `conneX` will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, `conneX` will automatically send you a confirmation email. If you do not receive such an email, your submission was not received. If you have problems with the submission process, send an email to the instructor before the due date.