

Lab 05

Deadline: 09/04/2022(11:59PM)

IIT Goa

Course Title: **Algorithm Design**

Course No: **CS 222**

In this assignment, you have to write a program that reads a pattern string P from the input, and a file *source.txt* containing n strings S_1, S_2, \dots, S_n (say, looking for a pattern in a given set of DNAs). The task in hand is to identify the presence of P in each S_i *with least expense*. You are supposed to start by writing a function *compChar*(S, P, j, k) which returns 0 if $S[j] = P[k]$ and 1 otherwise, where S and P are strings, and j and k are indices (integers). Whenever *compChar* returns 1, you are supposed to pay a penalty of 1\$ and nothing otherwise. Unfortunately, the pattern P and the strings are protected, in the sense that no function other than *compChar* does have the permission to read characters, neither in pattern P , nor the strings in the file provided with.

Your program should “print” n lines, where the i^{th} line starts with a Y/N (indicating whether P is present in S_i or not) followed by the minimum total penalty you paid while processing S_i . Implement *KMP matcher algorithm* to minimize the total penalty as much as possible. You may assume the pattern to be much much smaller compared to the strings in the file, so that the prefix computation cost associated with the KMP algorithm is negligible and hence doesn't appear as part of expense associated with any of those S_i s.

OR

Assume that you are staying in Andaman Islands and your home is in one of those islands. Islands are connected by boat services which you can avail by paying a fee where the return journey back home is free of cost (islands are small enough so that the travel cost within is negligible). Your objective is to design a travel assistant map sort of tool that will give you the least cost path from your home to any of those destination islands. You are being provided with a graph representing the information regarding the cost for commuting between islands. Use Dijkstra's algorithm to prune all those redundant/unnecessary information to set up your tool. The graph is provided in the file *graph.txt* where first line contains n (number of islands), second line m (number of boat services), and each of those m lines afterwards represent one of those boat services, with three fields, source, destination, and cost of journey.

You may assume islands named 1 to n where your home is in island 1. Once the graph being processed, your program should keep on prompting user for destinations and provide them with the shortest path, until the program being forcefully terminated. If the input is, say D , your output should be of the form:

The shortest path is: $1 - > \dots - > D$ of total cost c

and this computation should happen in $\mathcal{O}(k)$ time where k is the total number of boat services in this shortest path.

**Recommended reference is textbook (Cormen) but nothing else.

*****END*****