

**Indian Institute of Technology, Mandi**  
**August - November 2019**  
**CS202 - Data Structures and Algorithms**  
**Programming Assignment 3**

Course Instructor : Aditya Nigam  
Assignment created by : Ranjeet Ranjan Jha  
25 October 2019

## **Instructions**

- Plagiarism is strictly prohibited. In case of violation, an F grade will be awarded.
- Submit the complete code as cs202assignment3.zip file with a Makefile.
- For this assignment you can refer to Section 6.6 Sequence Alignment of the book Algorithm Design by Kleinberg and Tardos.
- The deadline for submission is **Friday, 1<sup>st</sup> November, 2019, 08:00 HRS (8:00 am)**. No late submissions will be entertained.
- Contact Abhigyan Khaund (9958980594), Ranjeet Ranjan Jha (7066896525) or Daksh Thapar (9592563214) for any queries.

# 1 Dynamic Programming

Solve the the following Dynamic Programming problem with optimization using C++ programming languag

1. The similarity between two strings where similarity of two strings X and Y is defined as M which is gap penalty or mismatch cost for pair of characters at same index in the two strings. Gap penalty and Mismatch cost is as defined in the book by Kleinberg and Tardos section 6.6.
2. You need to align two input DNA Sequences such that this cost of M for these two sequences is minimum.
3. Basic correct implementation of the algorithm - **(Marks-10)**
4. Optimization in Linear Space via Divide and Conquer **(Marks-20)**

*Above marks will be provided only if below mentioned files are present.*

You can implement these in two cpp files, one for first part and other for second part.

For the evaluation of the above files, each file should take two DNA strings as input and output the optimal alignment of the two strings and its cost as per the gap and mismatch cost.

## Output Format -

Optimal Alignment of String 1

Optimal Alignment of String 2

Cost = <Cost Value>

Use – to represent a gap in string alignment.

See sample I/O for more understanding

Compile the files to take an argument from the command line as **1** or **2**.

1. If argument is 1 - gap penalty = 6, mismatch penalty = 4

2. If argument is 2 - gap penalty = 4, mismatch penalty = 6

**Note:** We will run the code in given format -

**./part1 1**

**./part1 2**

**./part2 1**

**./part2 2**

## Sample Input Output:

argument 1-

**Input -**

ATAT

GCAT

**Output -**

ATAT

GCAT

Cost = 8

**argument 2 -**

**Input -**

ATAT

TATA

**Output -**

-ATAT

TATA-

Cost = 8

**IMPORTANT:** Do not hard code anything for any of the part. The argument change should only change the penalty values and should have no other change in the code.

**OPTIONAL (for practise only):** Take input an argument three and have different penalty values for A-G and T-C mismatch from the usual mismatch penalty.