CS205: Design and Analysis of Algorithms **Spring 2022**

Assignment 1

IIT Dharwad *Computer Science and Engineering*

Instructions:

- You are expected to solve this assignment manually and submit the relevant files in the required format.
- There are 2 questions in this document. Each carries 5 marks.
- Due on 21st January. With 25% penalty, you can submit up to 23rd January.
- There will be a penalty of 5% if you deviate from the naming scheme for any question - read the questions for more details.

Q1. We have discussed many variants of boolean satisfiability problems. One of the most important variant among them is 3-SAT, which is defined below.

3-SAT
Input: A formula Φ in Conjunctive Normal Form (CNF) such that each clause contains exactly three literals of distinct variables.
Question: Does there exist a truth assignment such that at least one literal per clause in Φ is TRUE?

We have learned that 3-SAT is NP-complete. Now, let us come up with a new problem involving boolean variables. For two positive integers p, q, we define the problem (p, q)-Boolean Set as follows:

(p, q)-Boolean Set
Input: A set Φ such that each element in Φ is a set (known as a clause) of p literals of distinct boolean variables.
Question: Does there exist a truth assignment of the boolean variables such that only at most q literals per clause in Φ are FALSE?

For example, $\{\{x_1, \overline{x_2}, \overline{x_3}\}, \{\overline{x_1}, \overline{x_3}, x_4\}, \{x_1, \overline{x_3}, \overline{x_5}\}, \{x_3, x_4, x_5\}\}$ is a yes-instance of $(3, 1)$-Boolean Set. This is because, an assignment of $\{x_1 \leftarrow T, x_2 \leftarrow F, x_3 \leftarrow F, x_4 \leftarrow T, x_5 \leftarrow T\}$ makes sure that each clause has only at most 1 FALSE literal. On the other hand, $\{\{x_1, \overline{x_2}, \overline{x_3}\}, \{\overline{x_1}, x_2, x_3\}\}$ is a no-instance of $(3, 1)$-Boolean Set. This is because, no truth assignment can make sure that only at most 1 literal per clause is FALSE.

Your task is the following: Each student will be given with two numbers p and q such that $p \geq 4$ and $q < p$ (See Appendix A). The task is to prove either that the problem (p, q)-Boolean Set is NP-complete or that the problem can be solved in polynomial-time. If you are proving the NP-completeness, then the polynomial-time reduction that you come up with must be from 3-SAT. The answer should be in a pdf file named <roll-no>.pdf where <roll-no> is your roll-no. The pdf can be prepared using any editor of your choice - it should not be a scanned document.

Q2. The objective of this question is to show the steps involved in an execution of the dynamic programming algorithm (let's call it DP1) discussed in Lecture 13 to find all-pair shortest path distances of a directed weighted graph. Each student is given with an input graph (see <roll-no>.txt) which is a directed graph with 5 vertices and with integer weights on edges ranging from 1 to 5. It is guaranteed that there are no self loop in the input graph. The format of the input file is as follows: First line contains 2 integers - the number of vertices (n) and the number of edges (m) respectively. The vertices are numbered from 1 to 5. Then there are m lines where each line represents one directed weighted edge. Each line representing a directed weighted edge is of the form "u v w" - this implies that there is a directed edge from u to v where the weight of the edge is w.

Your task is to create a file (manually) with name <roll-no>-out.txt (where <roll-no> is your roll-no.). The file must contain all matrices obtained in an execution of DP1 on the input graph. There will be a total of 4 matrices

$D^{(1)}, D^{(2)}, D^{(3)}$, and $D^{(4)}$ (See 25.1 in the text book and refer to Lecture 13). These matrices should be written in order, i.e., $D^{(i)}$ should come before $D^{(i+1)}$. For each matrix, each row should be written in a separate line. There should be white spaces between entries in a row. Infinity should be denoted by 'i' (without quotes). There should be a blank line in between two matrices. See attached examples.

Appendix

Roll no.	p	q	Roll no.	p	q	Roll no.	p	q
200010001	14	3	200010024	11	8	200010044	12	5
200010002	12	8	200010025	8	6	200010045	7	4
200010003	10	3	200010026	13	3	200010046	12	6
200010004	13	10	200010027	9	2	200010047	8	4
200010005	6	4	200010028	14	4	200010048	6	3
200010006	12	2	200010029	13	9	200010049	12	7
200010007	12	3	200010030	14	7	200010050	8	5
200010008	13	11	200010031	8	3	200010051	9	3
200010009	9	5	200010032	7	2	200010052	13	7
200010010	13	2	200010033	5	3	200010053	14	2
200010011	10	7	200010034	11	4	200010054	9	6
200010012	14	6	200010035	12	4	200010055	11	7
200010013	14	5	200010036	7	3	200010056	12	9
200010014	13	8	200010037	10	4	200010057	11	3
200010016	13	5	200010038	6	2	200020007	5	2
200010017	13	6	200010039	8	2	200020008	10	5
200010018	4	2	200010040	12	10	200020049	10	8
200010020	11	5	200010041	11	9	200030041	10	2
200010021	9	4	200010042	10	6	200030056	7	5
200010022	11	2	200010043	9	7	200030058	11	6
200010023	13	4						