INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Parameterized Algorithms: Class Test 1 2020-21

Date of Examination: 21 September 2020
Duration: 1.5 Hours
Full Marks: 25
Subject No: CS60083
Subject: Parameterized Algorithms
Department/Center/School: COMPUTER SCIENCE AND ENGINEERING

1. Prove that for a parameterized problem, there exists a kernelization algorithm if and only if there is an algorithm for the problem running in time $O(f(k) + poly(n))$ where f is some computable function.

   [5 Marks]

2. Questions 2 and 3 are regarding **FAST** and **FVST**. You have already seen the problem FAST. FVST is the vertex variant: given a tournament and an integer k, the objective is to determine if there is a set S of at most k vertices such that $G \setminus S$ is a DAG.
   Note that the vertices of a DAG have a topological ordering (a permutation of the vertices of the DAG) – an ordering such that every arc of the DAG starts at a vertex with a higher index and goes to a vertex with a lower index. When the DAG is also a tournament, then it has a unique topological ordering. Also, you can think of the vertex set $V(G) = 1, 2, ..., n$.

   (a) Show that in tournaments, there is a directed cycle if and only if there is a directed triangle.

   [5 Marks]

   (b) Give algorithms of running time $O^*(3^k)$ for both FAST and FVST.

   [5 Marks]

3. In this question, we will improve the FVST Algorithm:

   (a) Show the following – given a graph G on n vertices and 2 permutations of $V(G)$, find a longest common subsequence in polynomial time. **Hint**: Think of longest *increasing* subsequence.

   [2 Marks]

   (b) Suppose T is a tournament with the following properties: (i) $T \setminus v$ is a DAG with a unique topological ordering I, (ii) the vertex v does not participate in any triangles. Where can v be inserted in the topological ordering I?

   [2 Marks]

   (c) Suppose you can find LCS of two permutations of $V(G)$ in polynomial time. Use this as a subroutine to design an algorithm for FVST running in $O^*(2^k)$ time. **Hint**: Think of Iterative Compression.

   [6 Marks]

Best of luck