CMPT404/705: Design and Analysis of Computing Algorithms

Homework Assignment 3

Instructor: Igor Shinkar *Due date: March 18, 2022*

Question 1 (3 points) *Consider the following sequence*

- $a_0 = 0$.
- $a_1 = 1$.
- $a_n = \sum_{i=1}^{n-1} i \cdot a_i$ *for all* $n \geq 2$.

For example

- $a_2 = 1 \cdot a_1 = 1 \cdot 1 = 1$
- $a_3 = 1 \cdot a_1 + 2 \cdot a_2 = 1 \cdot 1 + 2 \cdot 1 = 3$

Write an algorithm that gets a number $n \geq 0$ *and returns* a_n *in time* $O(n^2)$.

Question 2 (3 points) *Write an algorithms that gets an* acyclic *directed graph* $G = (V, E)$, *and returns the length of the longest directed path in* G *in time* $O(|V| + |E|)$.

Question 3 (3 points) *Write a polynomial time algorithm for the Longest Common Subsequence problem. The input consists of two strings, and the goal is to find their longest common substring.*

For example, if the input strings are $x = ABCBDAFB$ *and* $y = BDCABAF$, *then their longest common substring is* $BCBAF$. *(The subsequences are underlined:* $x = A\underline{BCB}D\underline{AF}B$ *and* $y = \underline{B}D\underline{C}A\underline{BAF}$.*)*

Note that the substring doesn't need to be consecutive in the given strings.

Question 4 (3 points) *In class we saw a DP based algorithm for the decision version of the Hamiltonian Path problem. Modify the algorithm that given a graph* G *on* n *vertices runs in time* $O(2^n \text{poly}(n))$ *and returns a Hamiltonian Path in* G *(or reports "NO HamPath" if* G *does not have a hamiltonian path).*

Question 5 (3 points) *Recall that a* vertex cover *of a graph* $G = (V, E)$ *a subset of the vertices* $S \subseteq V$ *that touches all edges of the* G. *More formally, given an undirected graph* $G = (V, E)$, *a set* $S \subseteq V$ *is a vertex cover of* G *if for every* $e = (u, v) \in E$ *it holds that either* $v \in S$ *or* $u \in S$.

Suppose we have an algorithm ALG_{VC} *that gets an undirected graph* $G = (V, E)$ *with* $|V| = n$ *vertices and a parameter* k, *runs in* $\text{poly}(n)$ *time and outputs a vertex cover of* G *of size* k *(or reports that* G *does not have a vertex cover of size* k*).*

Design an algorithm ALG_{CLIQUE} *that gets an undirected graph* $G = (V, E)$ *with* $|V| = n$ *vertices and a parameter* ℓ, *runs in* $\text{poly}(n)$ *time and outputs a clique of size* ℓ *in* G *(or reports that* G *does not have a clique of size* ℓ*).*