University of Washington
Department of Computer Science and Engineering
CSE 421, Autumn 2016

November 14, 2016

Homework 8, Due Wednesday, November 23, 2016

Problem 1 (10 points):

Page 312, Exercise 1.

Problem 2 (10 points):

Give an algorithm, which given a directed graph $G = (V, E)$, with vertices $s, t \in V$ and an integer k, determines the number of paths from s to t of length k. Your algorithm should be polynomial in k, $|V|$ and $|E|$.

Problem 3 (10 points):

Give an $O(n^2)$ algorithm for finding the longest strictly increasing subsequence of a sequence of n integers. (Note that this problem can be solved in $O(n \log n)$ time by a non-dynamic programming style algorithm, but you do not need find it. Use of dynamic programming for this problem is recommended, but not required.)

Problem 4 (10 points):

The Chvatal-Sankoff constants are mathematical constants that describe the length of the longest common subsequences of random strings. Given parameters n and k, choose two length n strings A and B from the same k-symbol alphabet, with each character chosen uniformly at random. Let $\lambda_{n,k}$ be the random variable whose value is the length of the longest common subsequence of A and B. Let $E[\lambda_{n,k}]$ denote the expectation of $\lambda_{n,k}$. The Chvatal-Sankoff constant γ_k is defined at

$$\gamma_k = \lim_{n \to \infty} \frac{E[\lambda_{n,k}]}{n}.$$

Experimentally determine (by implementing an LCS algorithm), the smallest value of k, such that $\gamma_k < \frac{1}{2}$. In other words determine how large an alphabet needs to be so that the expected length of the LCS of two random strings is less than half the length of the strings.