COS 423	Theory of Algorithms	Spring 2018

Problem Set 5

This assignment is due Wednesday, April 4 at 11pm via electronic submission. Collaboration is permitted, according to the rules specified in the syllabus.

Read CHAPTER 5.5–5.6, 6.1–6.3 in *Algorithm Design*.

1. Fast matrix multiplication can be used to speed up a number of algorithms on dense graphs, including those for transitive closure, matchings, and shortest paths. In this problem, you will use fast matrix multiplication to find short cycles in a digraph.

   (a) Given a simple digraph $G = (V, E)$, design an algorithm to find a *triangle*—a directed cycle of length 3—or report that no such triangle exists. Your algorithm should take $O(n^{\log_2 7})$ time, where $n = |V|$.

   Hint: use the algorithm from precept that multiplies two n-by-n boolean matrices in $O(n^{\log_2 7})$ time (in the word RAM model).

   (b) Given a simple digraph $G = (V, E)$, design an algorithm to find a shortest directed cycle (or report that the digraph is acyclic). Your algorithm should take $O(n^{\log_2 7} \log^2 n)$ time.

   Hint: given an integer $k > 1$, design a subroutine that finds a directed cycle (not necessarily simple) of length $\leq k$ in $O(n^{\log_2 7} \log k)$ time.

2. An n-by-n *Hankel matrix* H is specified as a vector $h = (h_0, h_1, \ldots, h_{2n-2})$ of length $2n - 1$. Using 0-based indexing, element $H_{ij} = H_{ji} = h_{i+j}$, so that

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-2} & h_{n-1} \\ h_1 & h_2 & & \iddots & \iddots & h_n \\ h_2 & & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & & h_{2n-4} \\ h_{n-2} & \iddots & \iddots & & h_{2n-4} & h_{2n-3} \\ h_{n-1} & h_n & \cdots & h_{2n-4} & h_{2n-3} & h_{2n-2} \end{bmatrix}$$

   Design an algorithm to compute the matrix–vector product of an n-by-n Hankel matrix H and a vector x of length n using $O(n \log n)$ complex floating-point operations. Hankel matrices arise in physics, statistics, machine learning, and signal processing, including the solutions to differential equations.

3. You must sequentially load m pallets onto a train with n cars, e.g., pallets 1, 2, and 3 go onto car 1; pallets 4 and 5 go onto car 2; and so forth. Pallet i weights $w_i > 0$ kilograms and each car has a weight limit of $W > 0$ kilograms. You do not want to load the cars greedily (load as many pallets as you can onto each car until the next pallet exceeds the weight limit) because you may end up with an unbalanced train. Instead, your goal is to minimize the *variance* of the total weight loaded into the n cars. If the total weight loaded onto car j is x_j, then the variance σ^2 is given by:

$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}(x_j - \mu)^2, \text{ where } \mu = \frac{1}{n}\sum_{j=1}^{n} x_j$$

Design a dynamic programming algorithm to sequentially load the m pallets onto the n cars so as to minimize the *variance* of the weights (or report that no solution is possible). Your algorithm should take $O(m^2 n)$ time and use $O(mn)$ space.

Note: full credit also for $O(mn^2)$ time and $O(mn)$ space.