Questions

1. Given a complete graph $G_n$ on $n$ vertices, and real weights $w(u, v)$ on edges, the weight of a path (an ordered chain of vertices) is given by the sum of the weights of the edges in the path. A path is simple when no vertex is reused in the path. Our goal is to find the heaviest simple path (with maximum weight).

(a) Describe an ILP for solving the heaviest path problem. For extra credit, describe an LP formulation (no integer constraints) for the same problem, or give arguments for why that might be hard.

(b) Describe a Simulated annealing, or any other stochastic iterative optimization formulation for the same problem. Use pseudo-code, but be precise.

(c) Describe a greedy, or any fast heuristic for the problem. The final solution is not guaranteed to be optimal, but the algorithm must be guaranteed to finish ‘fast’.

2. Generate a random weighted graph $G_n$ with $n$ vertices as follows:

(a) Choose an ordered subset of $k = \min\{n, 2\lceil \log_2 n \rceil\}$ vertices $P = \{x_1, x_2, \ldots, x_k\}$, where $x_i \neq x_j$ for all $i, j$, and $1 \leq x_i \leq n$ for all $i$.

(b) For each $x_i, x_{i+1} \in P$ set $w(x_i, x_{i+1}) = 10$ with probability $\frac{1}{2}$, and $w(x_i, x_{i+1}) = -3$ otherwise.

(c) For each $1 \leq u \neq v \leq n$, s.t. $u, v$ are not consecutive in $P$, set $w(u, v)$ to be a random integer between $-10$ and $+3$.

Output examples of $G_n$ for $n = 3, 5, 10$.

3. Implement the algorithms from Q1(b) and Q1(c) to solve instances of $G_n$. Do not use any knowledge of the simulated data (such as edge weight values) in your code. Generate many instances of $G_n$ with increasing values of $n$ up to $n = 100000$ (or, whatever is possible on your computers). Run your code on these examples, and give scatter-plots of (a) the running time as a function of $n$ and, (b) the weight of the maximum path computed for each example.

4. Given a SNP matrix, and fraction $f$, design and implement an algorithm to identify a set $C$ of columns, such that there is a subset of at least $fn$ rows, that the matrix restricted to columns in $C$ is all ones, and that $|C|$ is maximized. Run your code on the matrix provided, for $f \in \{0.3, 0.5, 0.7\}$. [Note. While there is a similarity to the clique finding problem, we are talking about a SNP matrix here].