DISCRETE OPTIMIZATION: PROBLEM SET 2

Problem 1. Let A be the adjacency matrix of a graph G with vertices $\{v_1, v_2, \ldots, v_n\}$. (Recall that the trace of a matrix is the sum of the diagonal elements of the matrix.)

(1) What does the i, j entry of A^2 count? (explain)
(2) What does the i, j entry of A^3 count? (explain)
(3) What does trace(A^2) count? (explain)
(4) What does trace(A^3) count? (explain)

Problem 2. A *dominating set* in a graph G is a set of vertices D such that every vertex of G is either *in* D, or is a neighbor of a vertex in D. Finding a small dominating set has important applications in networks, such as monitoring or controlling with lowest cost.

(1) Write matrix inequalities (involving the adjacency matrix and any other convenient matrices) that are satisfied exactly when a $\{0, 1\}$ vector $\mathbf{x}$ indexed by the vertices of G represents a dominating set.
(2) Using the above, write an optimization problem that describes a minimal dominating set.

Problem 3. In a directed graph (without edge weights), prove that the minimum length of an $s - t$ path is the same as the minimum number of pairwise disjoint $s - t$ cuts. (Recall that an $s - t$ cut is the set of all edges that point out of some set $U \subset V$, with $s \in U$ and $t \notin U$.)

Problem 4. Prove that the function f described in Djikstra's algorithm gives the actual distances from s to each vertex v. (Hint: Once a vertex is taken out of U, the function never changes. So it's enough to show that $f(u)$ holds the right value for the vertex u selected...)

Problem 5. Show that the edge weights being non-negative is important for Djikstra's algorithm. That is, come up with and explain an example where the algorithm *doesn't* find the shortest paths when the edge weights are allowed to be negative.

Problem 6. Make a clear drawing of the graph described in the hiring problem (in particular with the edges labelled, and illustrate Djikstra's algorithm step-by-step on the graph, to find the optimal solution. A description of the problem follows.

It costs \$800 to train each employee. It costs \$1200 to let each employee go. It costs \$1600 a month to keep each extra employee over what is needed.

Month	Extra Employees Needed
1	$b_1 = 8$
2	$b_2 = 10$
3	$b_3 = 7$
4	$b_4 = 8$

A graph that describes this situation is

$$V = \{(i, x) | 1 \leq i \leq 4, b_i \leq x \leq 10\} \cup \{(0, 0), (5, 0)\}$$

$$A = \{((i, x), (i + 1, y)) \in V \times V | 0 \leq i \leq 4\}$$

Where the weight of edge $((i, x), (i + 1, y))$ is the cost to train or let go of the number of employees described ($8(y - x)$ when $y \geq x$, and $12(x - y)$ when $x > y$) plus the cost of keeping extra employees for the new month ($16(y - b_{i+1})$).