Question 1. (65 points) Let M be the matrix containing the edge costs for an n-vertex directed graph G: $M[i,i] = 0$ and, if $i \neq j$, then $M[i,j]$ contains the cost of the edge (i,j) if such an edge exists, and if no such edge exists then $M[i,j] = \infty$. Assume that n is a power of 2, i.e., $n = 2^q$ for some integer q.

For any pair of $n \times n$ matrices A and B, we use $A \circ B$ to denote the "$(\min, +)$ product" of those matrices, which is similar to the traditional matrix multiplication except that min now plays the role previously played by scalar addition, and scalar addition now plays the role previously played by scalar multiplication. In other words if $C = A \circ B$ then

$$C[i,j] = \min_{1 \leq k \leq n} (A[i,k] + B[k,j])$$

Of course, computing C from A and B can be done in $O(n^3)$ time.
We use the notation $M^{(2)}$ to denote $M \circ M$, we use $M^{(3)}$ to denote $M \circ M \circ M$, etc.

1. (20 points) Give an algorithm that, given M as input, computes $M^{(n)}$ by doing only $\log n$ of the "$\circ$" matrix multiplications.

2. (20 points) Give an algorithm that, given M and an integer k as input ($1 \leq k \leq n$), computes $M^{(k)}$ by doing at most $2 \log n - 1$ of the "$\circ$" matrix multiplications.

   Hint. Use the binary representation of k, suppose it is $b_q b_{q-1} \cdots b_0$ where b_q is the most significant bit and $q = \log n$. For example if $k = 10$ then the binary representation is 1010, and you use the fact that $k = 0 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3$.

3. (25 points) Give an algorithm that, given M as input, computes $M^{(n)}$ in $O(n^3)$ time.

   Hint. First try to come up with a graph-theoretic interpretation of $M^{(k)}[i,j]$, that is, what $M^{(k)}[i,j]$ means in terms of the graph G.

Question 2. (35 points) Let G and M be as in the previous question, except that G is undirected and the "$\circ$" product is now a $(\min, \max)$ product rather than a $(\min, +)$ one. That is, if $C = A \circ B$ then

$$C[i,j] = \min_{1 \leq k \leq n} (\max\{A[i,k], B[k,j]\})$$

Give an algorithm that, given M as input, computes $M^{(n)}$ in $O(e \log n + n^2)$ time where e is the number of edges in G.

Hint. First try to come up with a graph-theoretic interpretation of $M^{(k)}[i,j]$, that is, what $M^{(k)}[i,j]$ means in terms of the graph G.

Date due: November 8, 2012