

This assignment will not be collected. You can view it as extra practice for our final exam.

A. There are two ways to compute large powers of things (be they matrices, real numbers, integers modulo some modulus m , or whatever). The naive method computes A^k by multiplying A with itself k times. The repeated squaring method, which is discussed in our KHR textbook, instead computes powers A^{2^j} and then assembles A^k from them. Be sure to understand the running times of these two algorithms (where the basic operation is addition and multiplication of whatever kind is relevant).

B. In class, we've discussed an algorithm for testing whether a graph with adjacency matrix A was connected, by computing $A^0 + A^{n-2} + A^{n-1}$ and testing whether all of its entries were positive. What is the running time of this algorithm? The answer depends on which algorithm from Problem A you use. It also depends on which matrix multiplication algorithm you use (Assignment H). Be sure to understand all possible combinations.

C. What is the running time of breadth-first search? Figure it out from the class notes on our course web site. Check your answer against Section 11.3 of the DLN text; he might have a tighter bound.

D. Describe an algorithm that uses breadth-first search, instead of the adjacency matrix A , to determine whether or not an undirected graph is connected. What is the running time of the algorithm?

E. You should be able to execute Dijkstra's algorithm on a small graph. And what is its running time? See the class notes on our course web site, and Section 11.5 of the DLN textbook.