Let $f : \mathbb{N} \to \mathbb{N}$ be a function that grows without bound. That is, $\lim_{n\to\infty} f(n) = \infty$. We say that $g(n)$ is $\mathcal{O}(2^{f(n)})$ if there exist positive constants C, N such that $g(n) \leq C2^{f(n)}$ for all $n \geq N$. Similarly, we say that $g(n)$ is $2^{\mathcal{O}(f(n))}$ if there exist positive constants C, N such that $g(n) \leq 2^{Cf(n)}$ for all $n \geq N$.

A.A. Prove that if g is $\mathcal{O}(2^{f(n)})$ then g is $2^{\mathcal{O}(f(n))}$.

A.B. Find a g in $2^{\mathcal{O}(f(n))}$ that is not in $\mathcal{O}(2^{f(n)})$.

Earlier in our course — maybe on Day 15? — we described a Turing machine for testing whether a given directed graph G was in fact a connected undirected graph. For the sake of consistency, assume that $\langle G \rangle$ is (a reasonable version of) the adjacency matrix of G.

B. What are the time complexity and space complexity of that Turing machine? Analyze them in detail, and state your answers using $\mathcal{O}$ notation. Actually, give two answers for each: one in terms of the input size n, and one in terms of the number m of nodes in the graph.