This paper investigates how to efficiently and locally linearize graphs - i.e., how to build a sorted list of the nodes of a connected graph in a distributed and self-stabilizing manner. This problem has many interesting application domains; for instance, self-stabilizing algorithms for graph linearization can serve as a building block to construct robust peer-to-peer overlays. A foremost question addressed in this paper is how to measure the efficiency of a given algorithm. We introduce a new model that takes into account the parallel complexity of a protocol. Our model avoids the scalability problems and bottlenecks of existing frameworks. We also propose two variants of a simple, local linearization algorithm. For each of these variants, we present extensive formal analyses of their worst-case and best-case parallel time complexities, as well as their performance under a greedy selection of the actions to be executed. In particular, we show that one of the proposed algorithms achieves near-optimal parallel time complexity under such a greedy selection. We validate the behavior of these algorithms by experiments which confirm our formal findings and indicate that the runtimes may in fact be better in practice.