Discovering dense subgraphs and understanding their relations is important. Peeling algorithms (k-core, k-truss, and nucleus decomposition) have been shown to be effective to locate many dense subgraphs. However, constructing the hierarchy and even correctly computing the connected k- cores and k-trusses are mostly overlooked in the literature. Finding k-cores, k-trusses and nuclei, and constructing the hierarchy requires an additional traversal operation which is as expensive as the peeling process. In this work, we first fix the mistake in the literature by a thorough review of history, and then propose efficient and generic algorithms to construct the hierarchy of dense subgraphs for k-core, k- truss or any nucleus decomposition. Our algorithms leverage the disjoint-set forest data structure to efficiently construct the hierarchy during traversal. Furthermore, we introduce a new idea to get rid of the traversal. We construct the sub- graphs while visiting neighborhoods in the peeling process, and build the relations to previously constructed subgraphs. We also bring out an existing idea for k-core hierarchy, and adapted to our objective efficiently. Experiments on different types of large scale real-world networks show significant speedups over naive algorithms and existing alternatives.