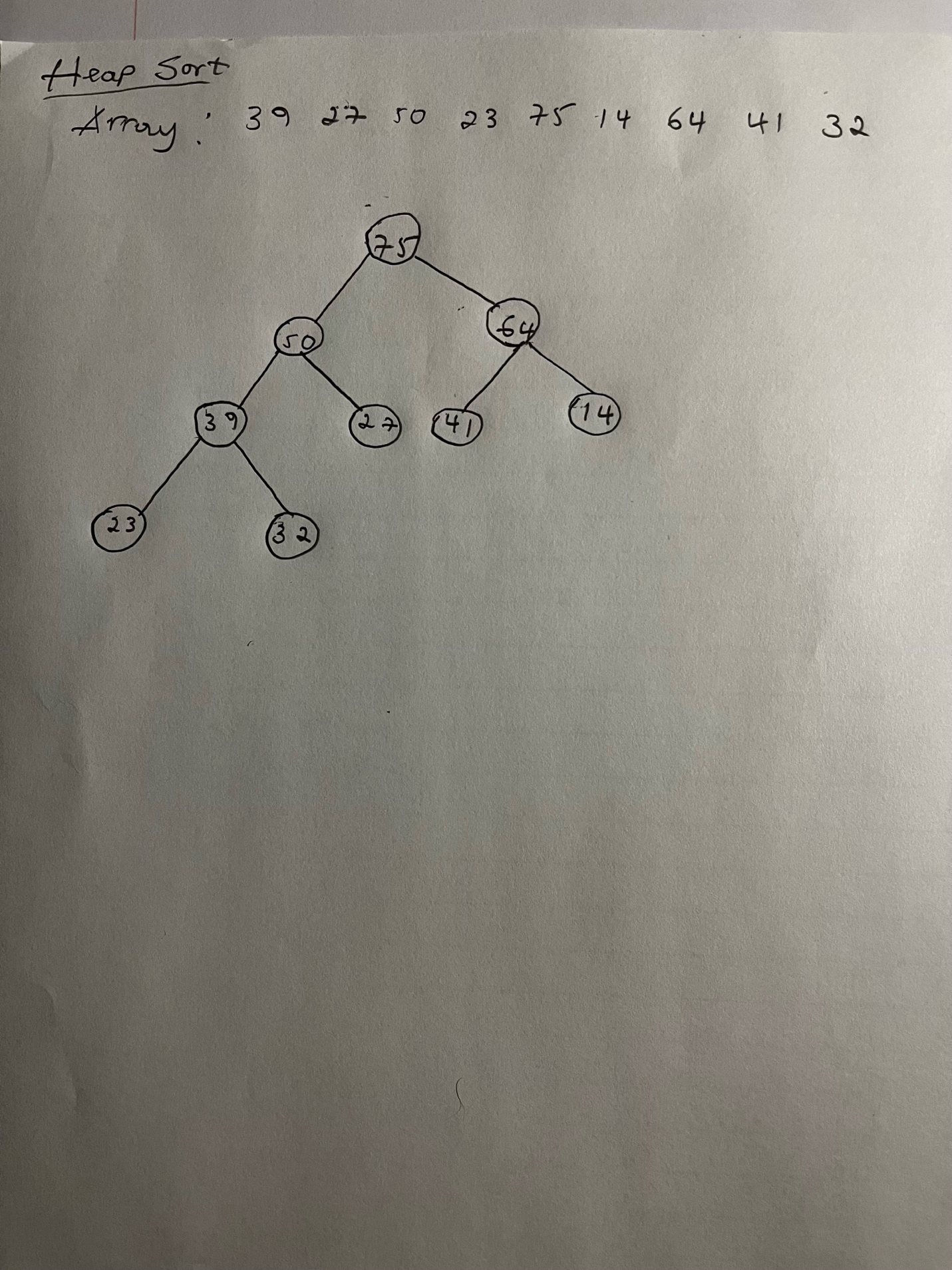
A piece of paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated



In testing the sorting times, I noticed that Quick Sort and Merge Sort were way ahead, especially for huge amounts of data, because they break the list into smaller pieces, sort those, and then put them all back together. This method, which is called O(NlogN), makes sorting big lists much faster. Quick Sort usually finishes first since it sorts directly in the list and cleverly picks where to start splitting it, although sometimes how well it does depends on these choices. Merge Sort is always steady, sorting at the same speed no matter how scrambled the list is at the start. So, for big lists, Quick Sort is often my go-to for speed, but Merge Sort is a strong backup for when I need sure results.

Insertion Sort, with its O(N^2) efficiency, is best left for smaller lists or ones that are almost sorted already. It's great when the list is short or doesn't need much fixing because it's straightforward and doesn't need much extra space. But when it comes to the big numbers I tested, Quick Sort was generally the fastest, provided the right starting points are chosen. Merge Sort was a solid plan B, reliable for any situation where I couldn't risk Quick Sort's variability. This testing really showed me the importance of picking the right tool for the job based on how much data I need to sort and what kind of order it's in to start with.