1. Specification

Given an array A of $n = 4^k$ (for some $k \in \mathbb{N}$) elements, sort it as follows:

- If $n \leq 4$, sort A with insertion sort and finish.
- Sort the first $\frac{3n}{4}$ elements recursively.
- Sort the last $\frac{3n}{4}$ elements recursively.
- Sort the first $\frac{3n}{4}$ elements recursively.

2. Proof

The algorithm deals with three-quarter-portions of A. We first prove that each such portion is sorted correctly. Then, we prove that by sorting each portion in the specified order, the array is sorted.

2.1. Each portion is sorted

The proof is by induction.

Define $P(k)$: an array of size 4^k or $3 \cdot 4^k$ is sorted correctly. The base case is for $k = 0$, i.e. $n = 1$ or $n = 3$. As $n \leq 4$, the correctness follows from the correctness of insertion sort.

Assume that $P(k)$ holds for $k = p$. Then, for $k = p + 1$, each portion is of length

$$\frac{3n}{4} = \frac{3 \cdot 4^{p+1}}{4} = 3 \cdot 4^p,$$

and is sorted correctly according to the inductive hypothesis.

The proof follows from the principle of induction.

2.2. The array is sorted

Let A_i be the i:th quarter-portion of A with length $\frac{n}{4}$, and a_i the set of the i:th smallest $\frac{n}{4}$ elements.

Consider an element $x \in A$. Initially, it might be in any A_i. Note that once x has been placed in the correct portion, it will never be moved to another one.

- If $x \in a_1$ or $x \in a_2$, the second sort ensures it is not in A_4, and as such it is placed in the correct portion in the third sort.
- If $x \in a_3$ or $x \in a_4$, the first sort ensures it is not in A_1, and as such it is placed in the correct portion in the second sort.

3. Complexity