2.1

A screen shot of a computer

Description automatically generated

2.2

A computer screen with white text

Description automatically generated

Part 2:

partial\_sort is more efficient than full sort because it only sorts elements up to the median position. We only need the middle elements to be in their correct sorted positions and don't care about the relative order of elements before or after the median. This saves work compared to fully sorting every single element into its exact position.

Part 3

nth\_element is even more efficient and is actually the most efficient of the three, with O(n) average time complexity. It's faster because it only ensures that elements before the nth position are smaller than the nth element and elements after the nth position are larger than the nth element. It doesn't sort any portions of the array. it just makes sure they're on the right side and doesn't care about their order at all. This minimal work makes nth\_element O(n) on average instead of O(n log n), which is a significant improvement. For median calculation, this is perfect since we only need the middle element(s) to be correct, and don't care about the order of any other elements.

2.3

A screen shot of a computer

Description automatically generated

The class-oriented version makes the code better because the code is more structured - data and related functions are grouped together in a logical way. Once we create a triangle object, we can easily reuse it, access its values, or print it multiple times without recalculating. In addition, data is protected by the private key word and can only be accessed through controlled methods, preventing accidental modifications.