Sort function how it works:

From the C++11/14 standard, std::sort is guaranteed to have:

**§25.4.1.1/3**

Complexity: O(N log(N)) (where N == last - first) comparisons.

The other, stable, standard sorting algorithm (namely std::stable\_sort) is guaranteed to have:

**25.4.1.2/3**

Complexity: It does at most N log²(N) (where N == last - first) comparisons; if enough extra memory is available, it is N log(N).

For std::forward\_list::stable, instead:

**23.3.4.6/26**

Complexity: Approximately N log(N) comparisons, where N is distance(begin(), end()).

The same goes for std::list:

**23.3.5.5/31**

Complexity: Approximately N log(N) comparisons, where N == size().

Binary search:

<https://www.geeksforgeeks.org/binary-search-algorithms-the-c-standard-template-library-stl/>

The time complexity of the binary search algorithm is **O(log n)**. The best-case time complexity would be O(1) when the central index would directly match the desired value. The worst-case scenario could be the values at either extremity of the list or values not in the list.

Valarray in c++:

https://www.geeksforgeeks.org/std-valarray-class-c/