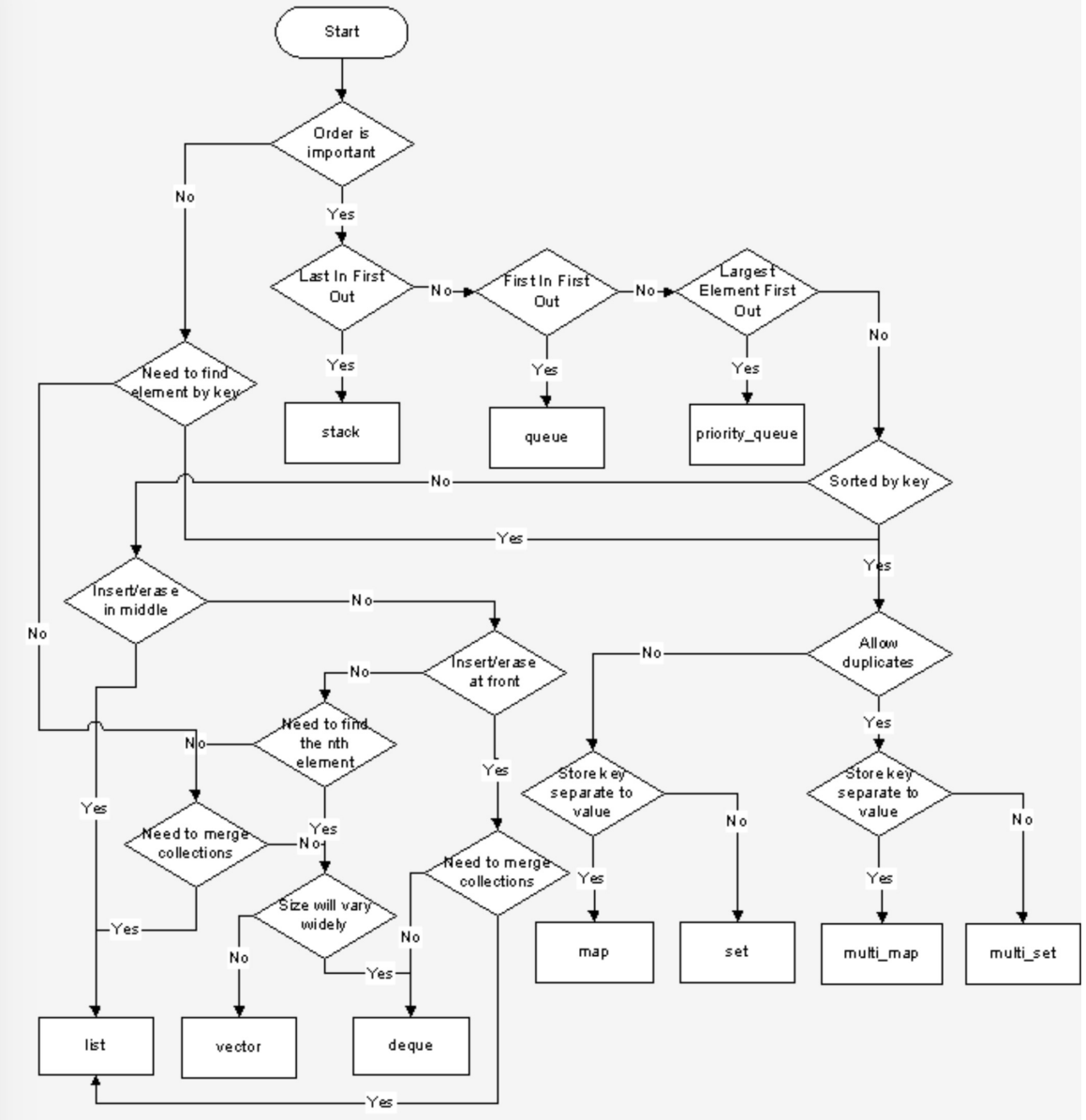
DATA STRUCTURES IN C++

Data Structure Decision Diagram



- The following diagram gives you the direction to which data structure to use in C++ according to the problem you are trying to solve

Note: I don't have the source of this diagram. If you know it, please drop me a msg so I can add it here.

Arrays

- Fixed-size collection of elements of the same type
- Stored in **contiguous memory**
- Declared with syntax: **`type arrayName[size]`**

  Example:

  `int numbers[5]`
- Can also be initialized at declaration:

  `int arr[3] = {1 ,2, 3}`
- Cannot resize after declaration
- Size can be calculated by **`sizeoff(arr) / sizeof(arr[0])`**
- stdlib provides **`std::array<type, size>`**
- **Example:**

  `std::array<int, 3> a = {1, 2, 3};`

*https://en.cppreference.com/w/cpp/container/vector

Arrays (vectors)

- std::vector is a sequence container that encapsulates dynamic sized arrays*

*https://en.cppreference.com/w/cpp/container/vector

Linked List

- desc

Stack

- desc

*https://en.cppreference.com/w/cpp/container/vector

Queue

- desc

*https://en.cppreference.com/w/cpp/container/vector

Heap

- desc

Hash Table

- desc

Tree

- desc