

42 Lecture - CS304

Important Mcqs

1. Which of the following is NOT a type of iterator in C++?

- a) Forward iterator
- b) Backward iterator
- c) Input iterator
- d) Random access iterator

Answer: b) Backward iterator

Which of the following is a feature of a forward iterator?

- a) Can move in both directions
- b) Can be used to modify the elements in a container
- c) Can access the elements of a container multiple times
- d) Can only move forward in a container

Answer: d) Can only move forward in a container

Which of the following is an example of a container that supports random access iterators?

- a) Linked list
- b) Queue
- c) Array
- d) Set

Answer: c) Array

What is the purpose of an output iterator?

- a) To iterate through a container in reverse order
- b) To modify the elements in a container
- c) To read the elements in a container
- d) To write data to a container

Answer: d) To write data to a container

Which of the following is an example of a bidirectional iterator?

- a) Stack
- b) Deque
- c) Forward list
- d) Map

Answer: b) Deque

What is the complexity of the operator++() function for a random access iterator?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(1)$
- d) $O(n^2)$

Answer: c) $O(1)$

Which of the following algorithms require a random access iterator?

- a) `std::sort()`

- b) `std::transform()`
- c) `std::reverse()`
- d) `std::unique()`

Answer: a) `std::sort()`

Which of the following is a characteristic of an input iterator?

- a) Can be used to modify the elements in a container
- b) Can only access the elements of a container once
- c) Can move in both directions
- d) Can skip elements in a container

Answer: b) Can only access the elements of a container once

Which of the following is an example of a container that supports bidirectional iterators?

- a) Hash table
- b) Binary search tree
- c) Vector
- d) Queue

Answer: b) Binary search tree

Which of the following is an example of a container that supports forward iterators?

- a) Stack
- b) Map
- c) Linked list
- d) Set

Answer: c) Linked list