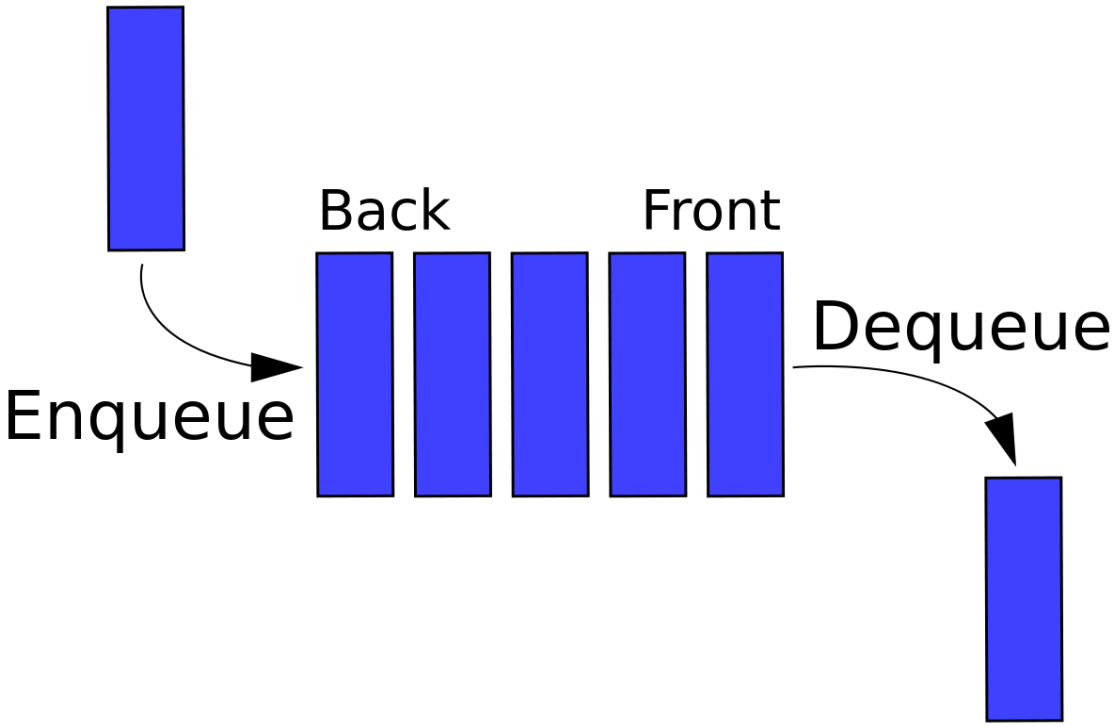
Queue

“...using Single Linked List”

Prerequisite: Single Linked List

Md. Saidul Hoque Anik
onix.hoque.mist@gmail.com

Properties of Queue



Application of Queue

1. Any kind of line in booth (Bank, shopping mall, hospital etc.)
2. Scheduling task
3. Printer Jobs
4. Keyboard buffer
5. One way Traffic

Properties of Queue

1. Non-primitive Linear (Sequential) Data Structure
2. Insert/Delete Behavior : FIFO
3. Supported operations are-
    i. Enqueue
    ii. Dequeue
    iii. isEmpty
    iv. isFull
    v. makeEmpty

Operations

Time complexity in big O notation

Algorithm	Average	Worst case
Space	O(*n*)	O(*n*)
Search	O(*n*)	O(*n*)
Enqueue	O(1)	O(1)
Dequeue	O(1)	O(1)

Variation of Queue

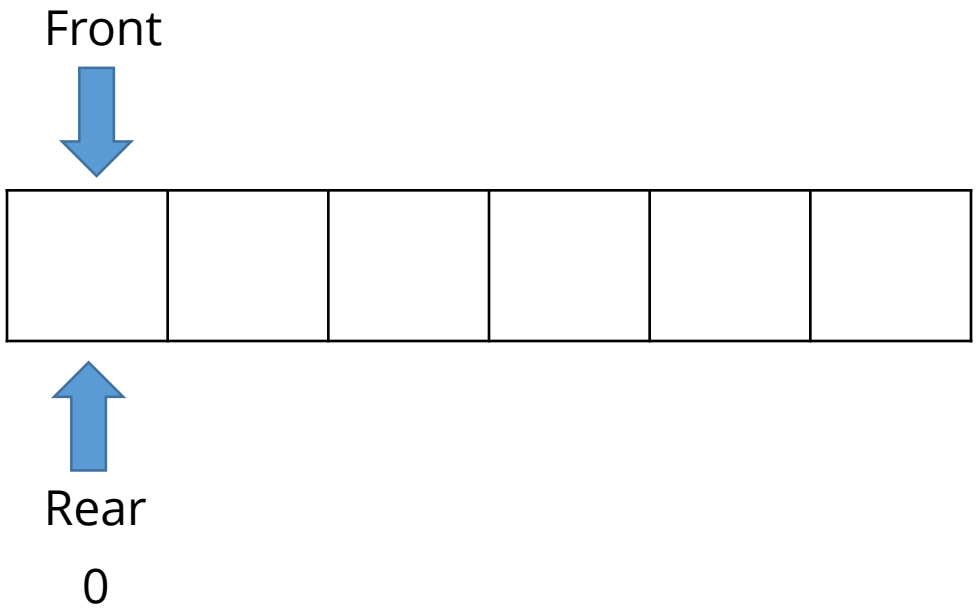
1. Linear Queue
2. Circular Queue (Also known as **Ring Buffer**)
3. Double Ended Queue
4. Priority Queue

Implementation of Queue

1. Using Array
2. Using Linked List

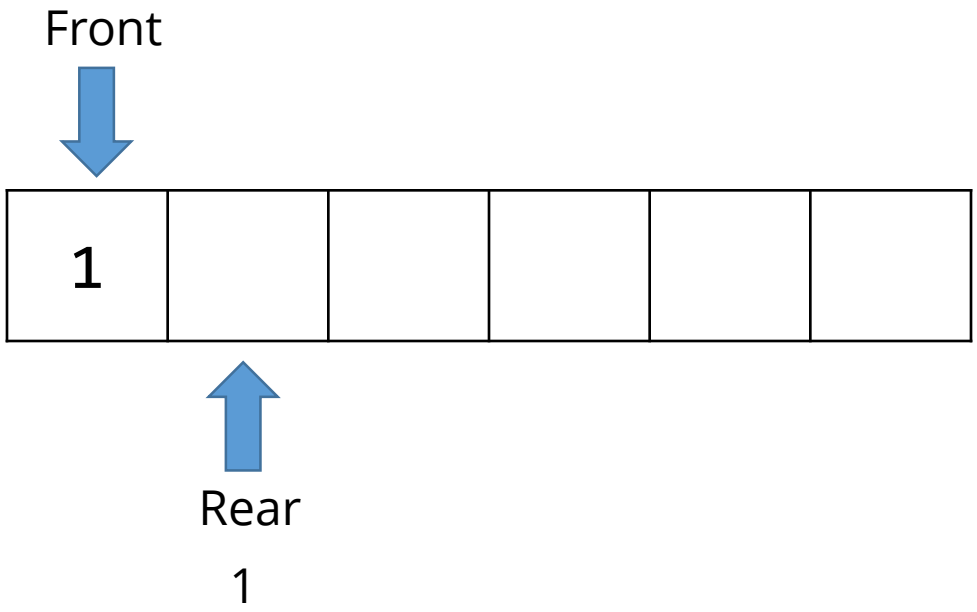
Implementation Using Array

Capacity = 6
Length = 0



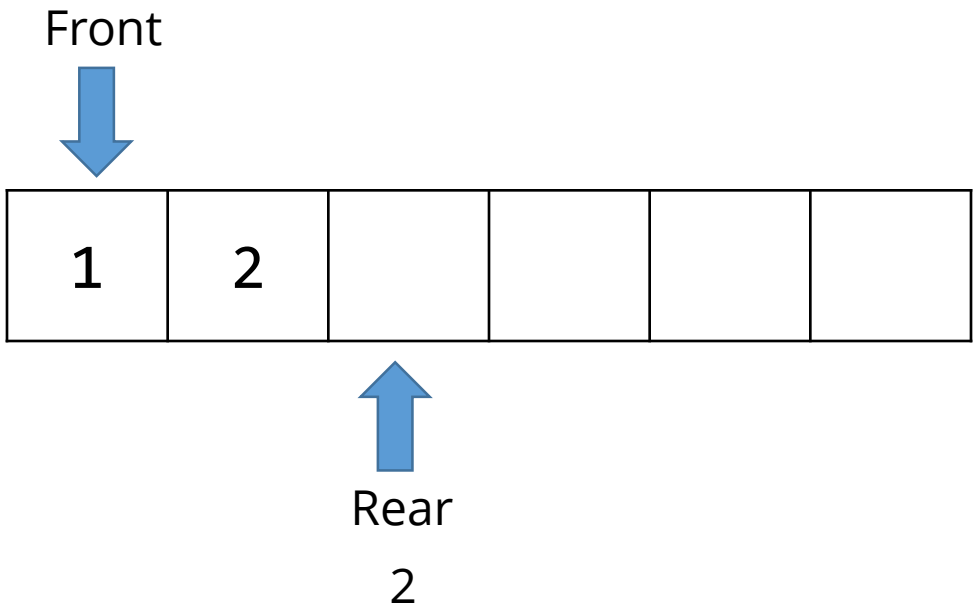
Implementation Using Array

Capacity = 6
Length = 1



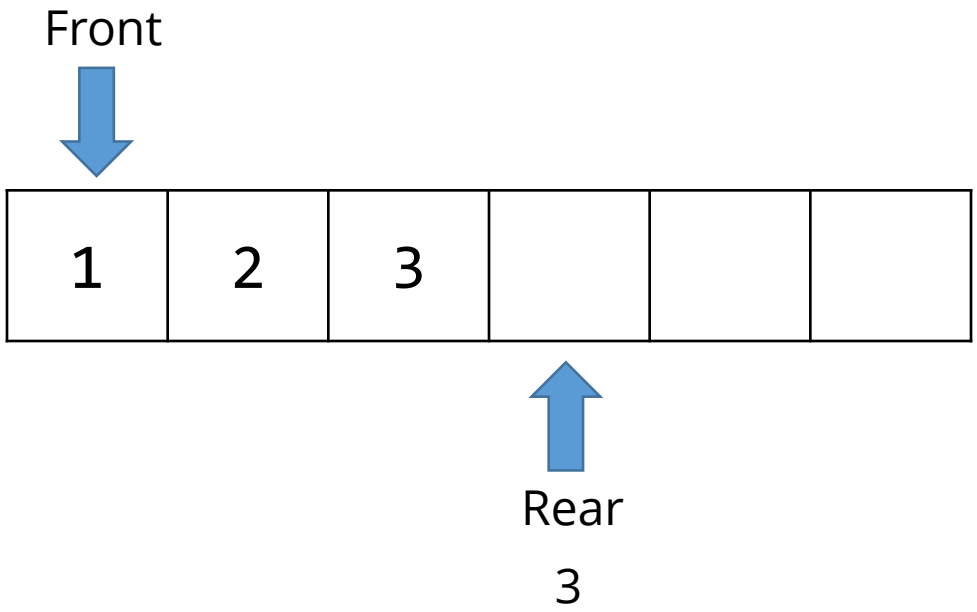
Implementation Using Array

Capacity = 6
Length = 2



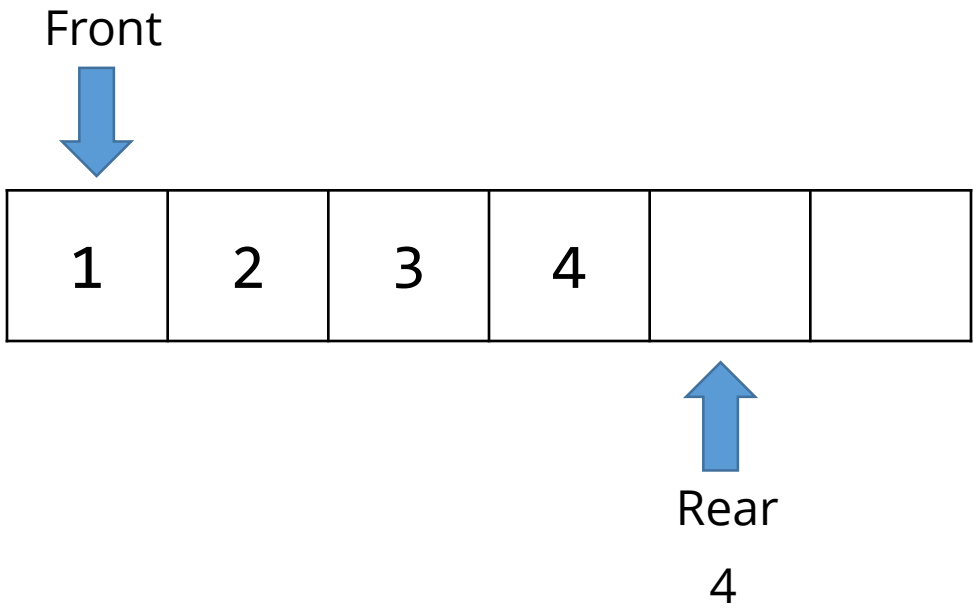
Implementation Using Array

Capacity = 6
Length = 3



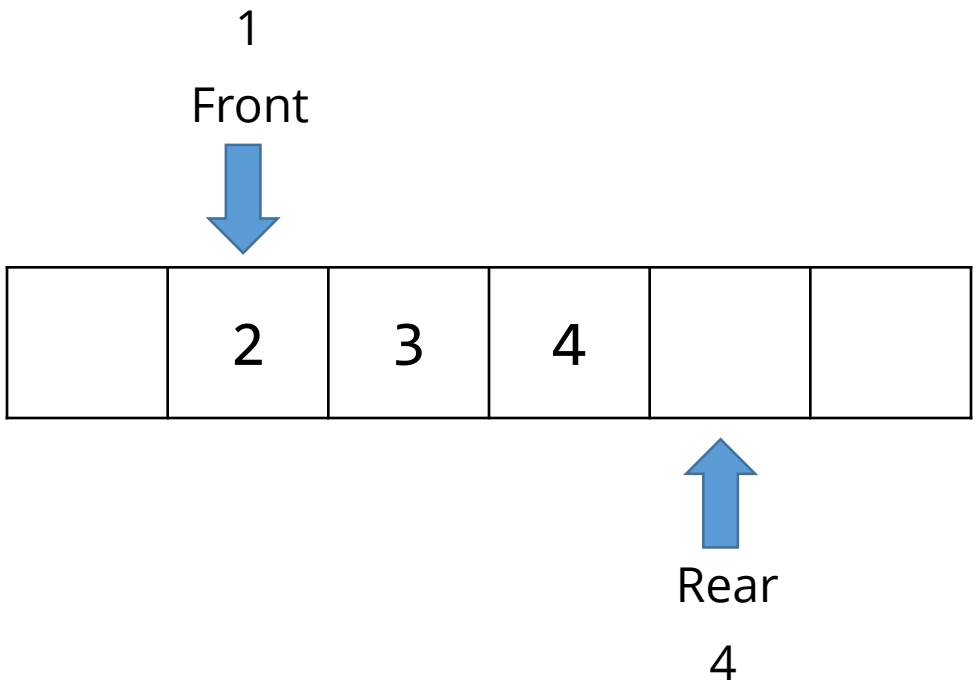
Implementation Using Array

Capacity = 6
Length = 4



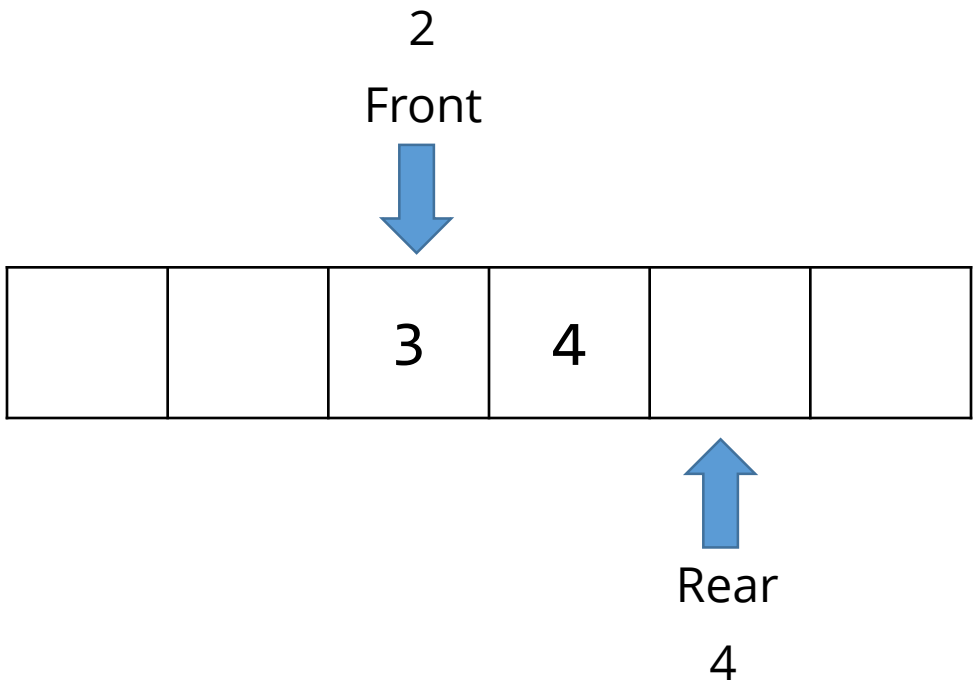
Implementation Using Array

Capacity = 6
Length = 3



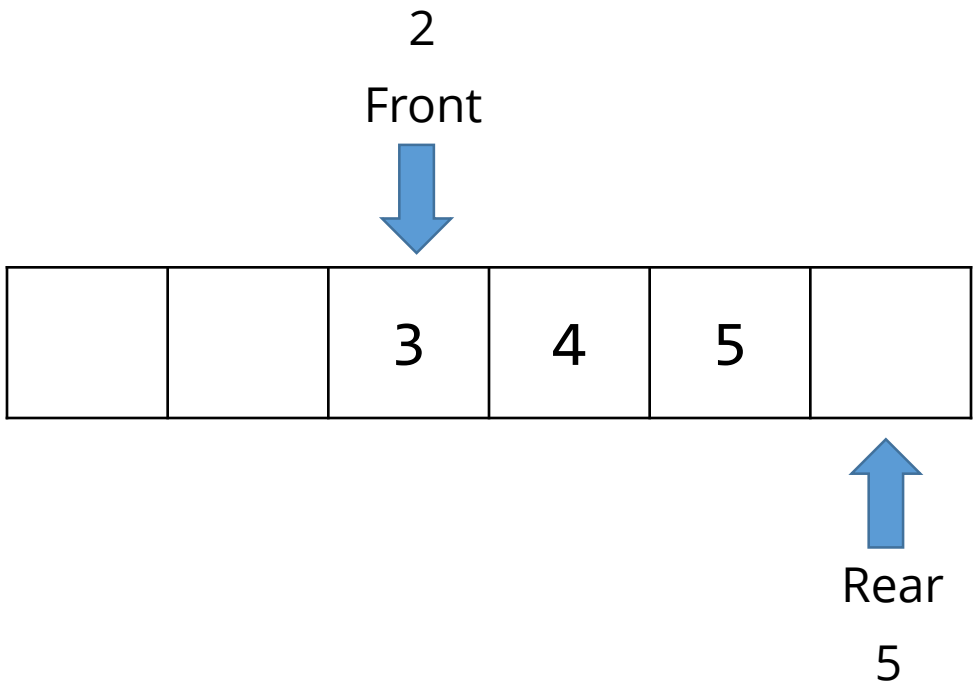
Implementation Using Array

Capacity = 6
Length = 2



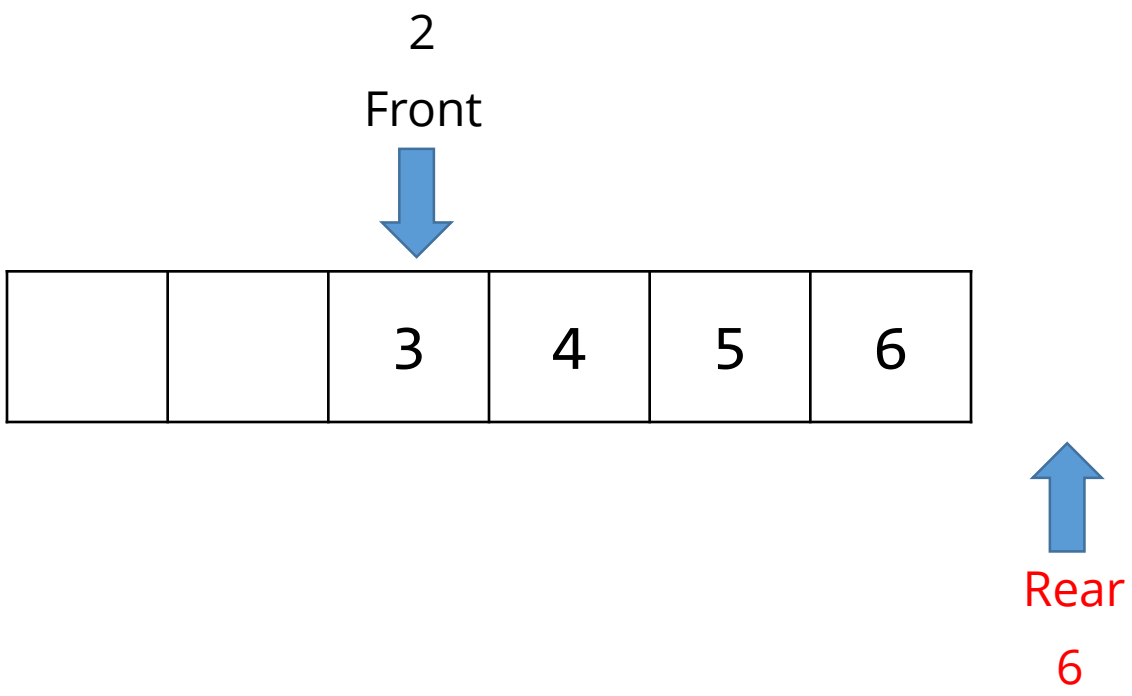
Implementation Using Array

Capacity = 6
Length = 3



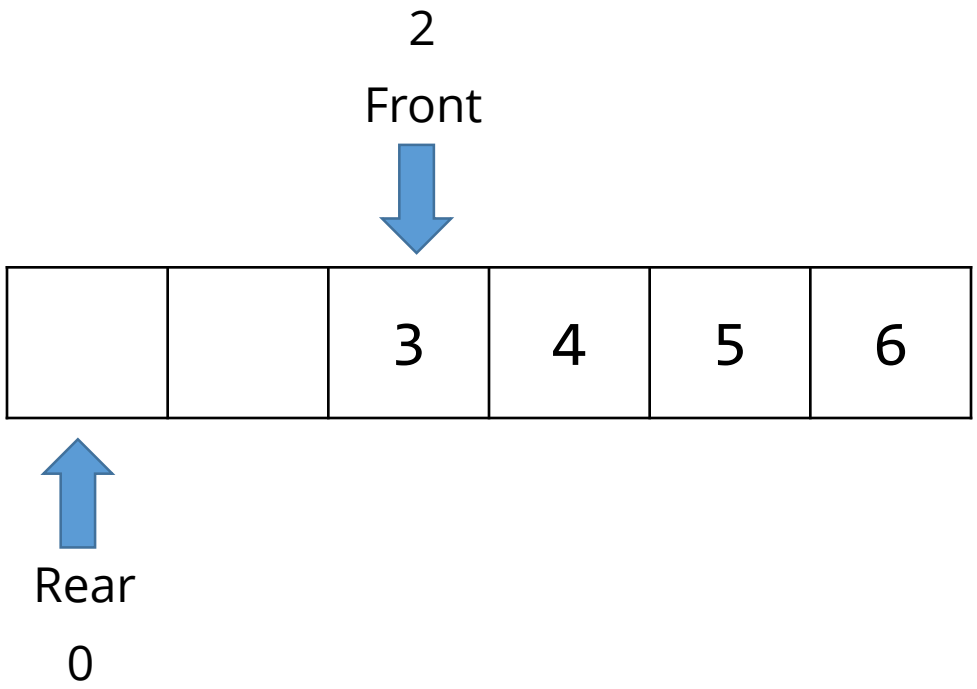
Implementation Using Array

Capacity = 6
Length = 4



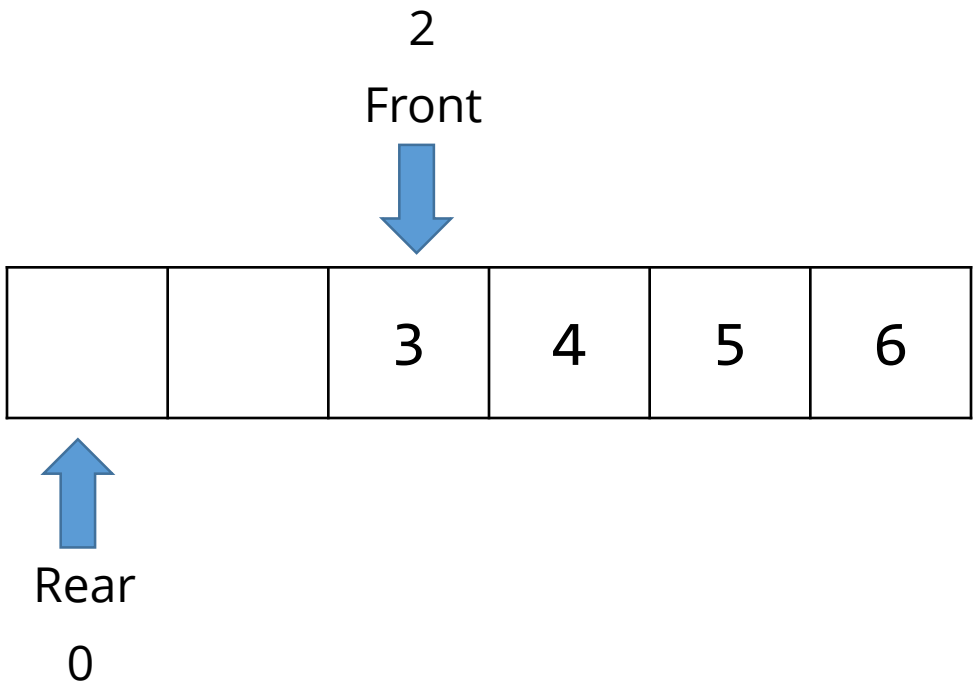
Circular Queue

Capacity = 6
Length = 4



Circular Queue

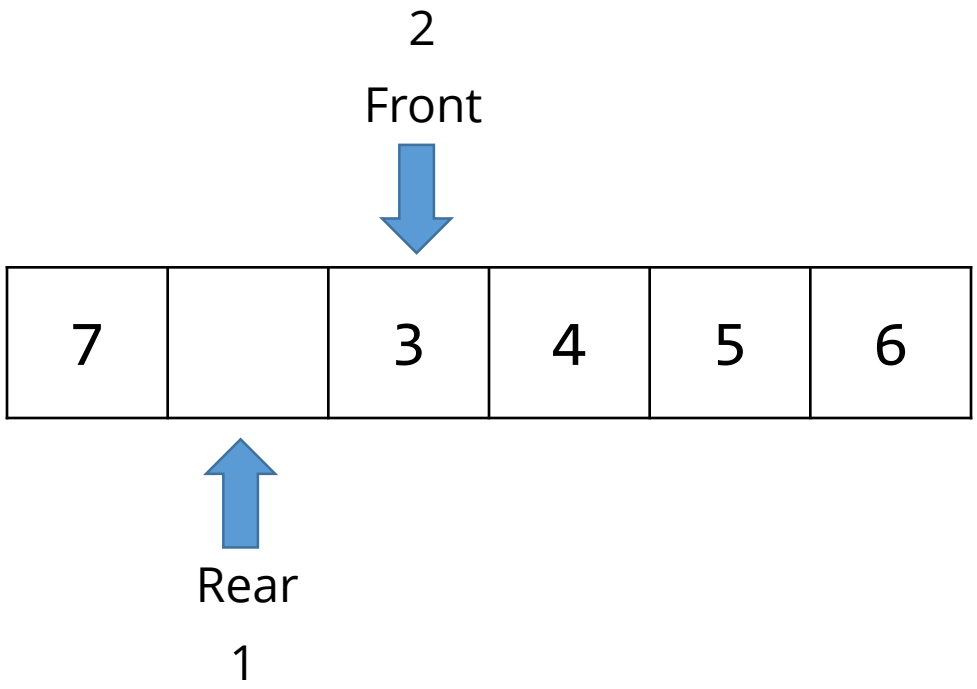
Capacity = 6
Length = 4



Rear = (Rear + 1) mod capacity

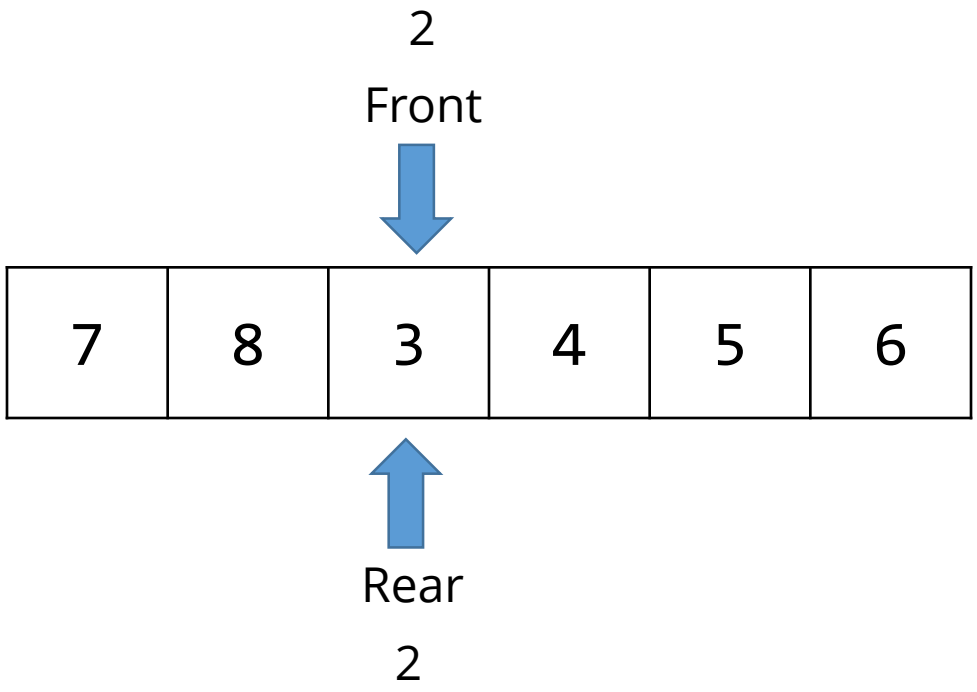
Circular Queue

Capacity = 6
Length = 5



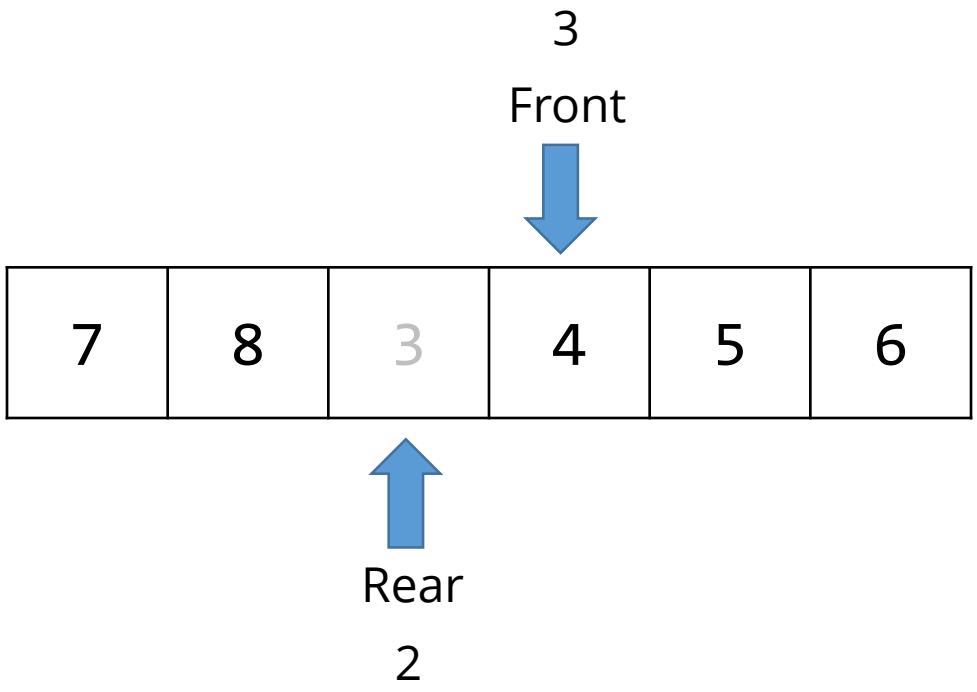
Circular Queue

Capacity = 6
Length = 6



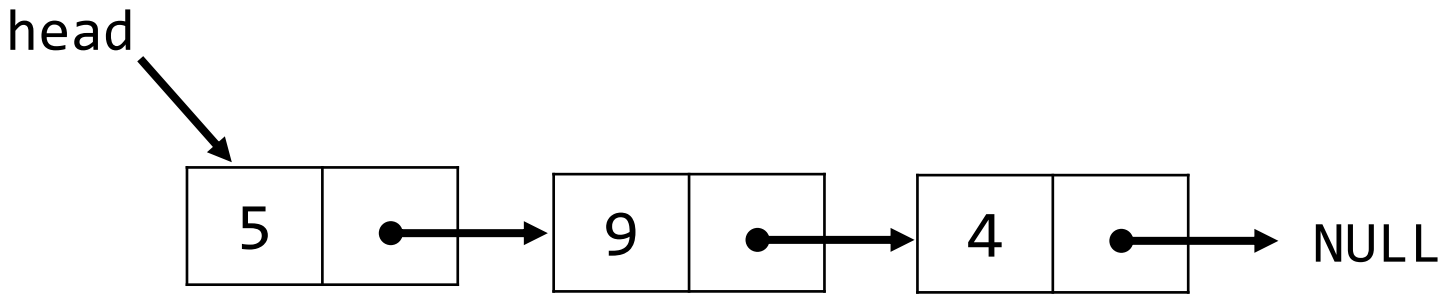
Circular Queue

Capacity = 6
Length = 5



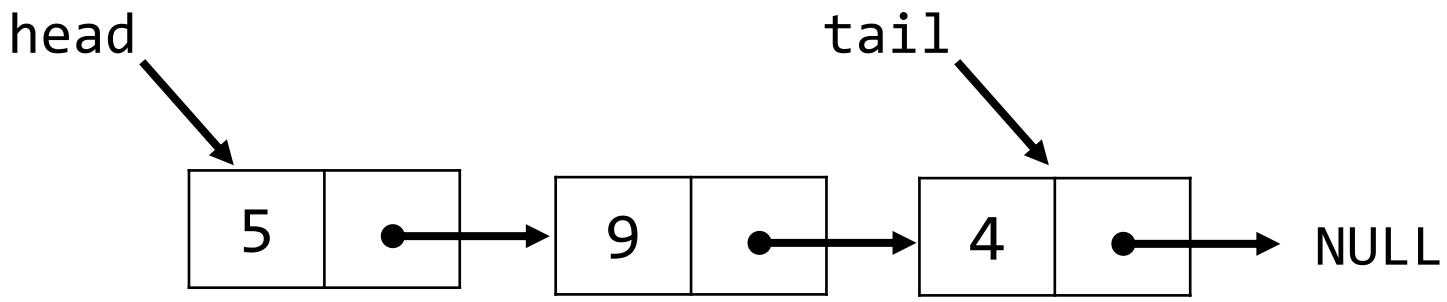
Implementation Using Linked List

Implementation Using Linked List



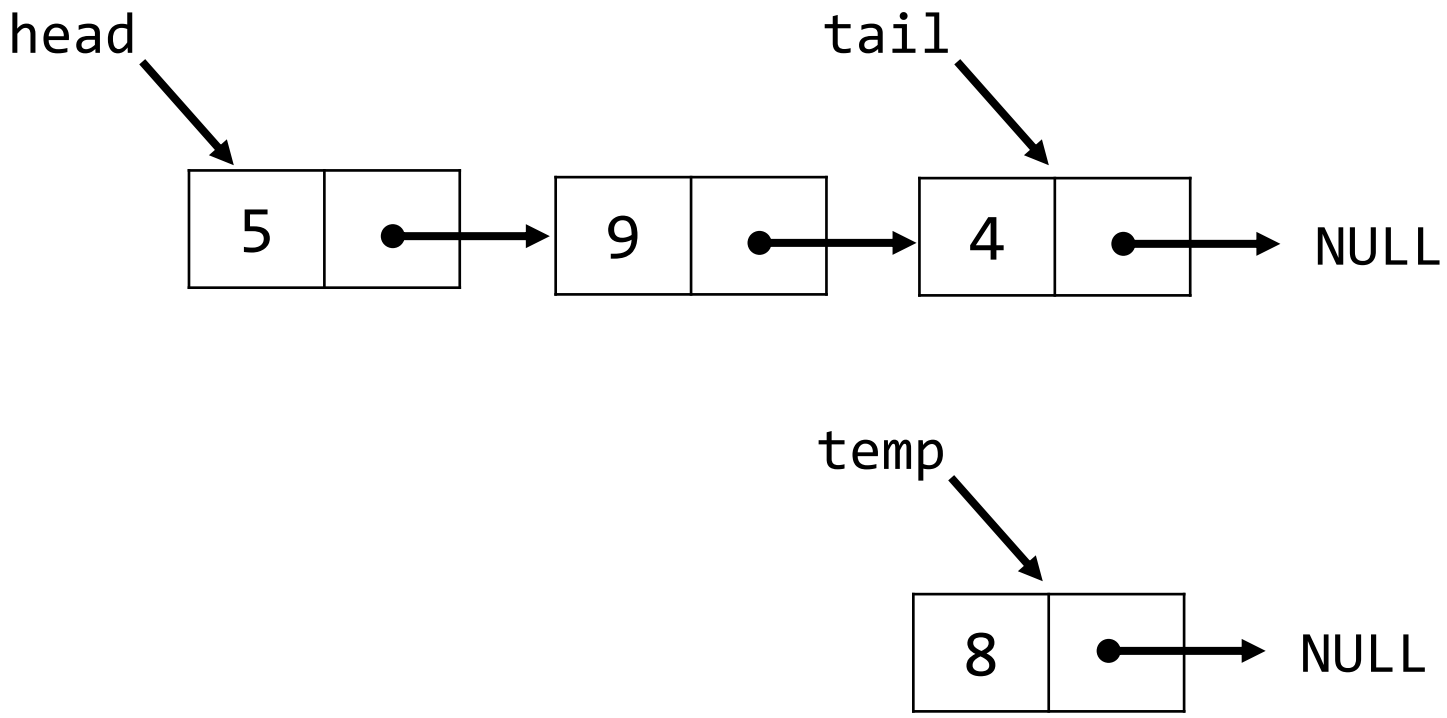
Implementation Using Linked List

Dequeue = LinkedList.pop_front()
Enqueue = LinkedList.push_back()



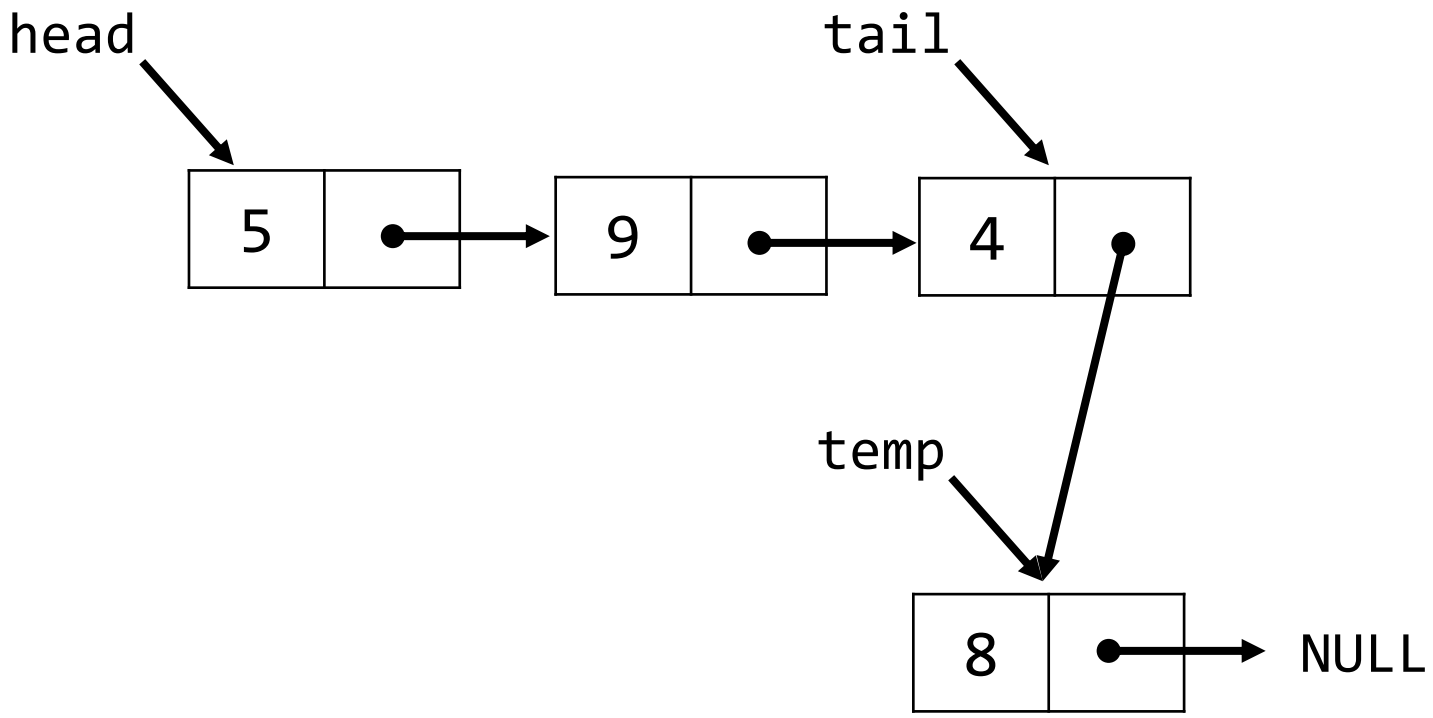
Implementation Using Linked List

Push_back Implementation



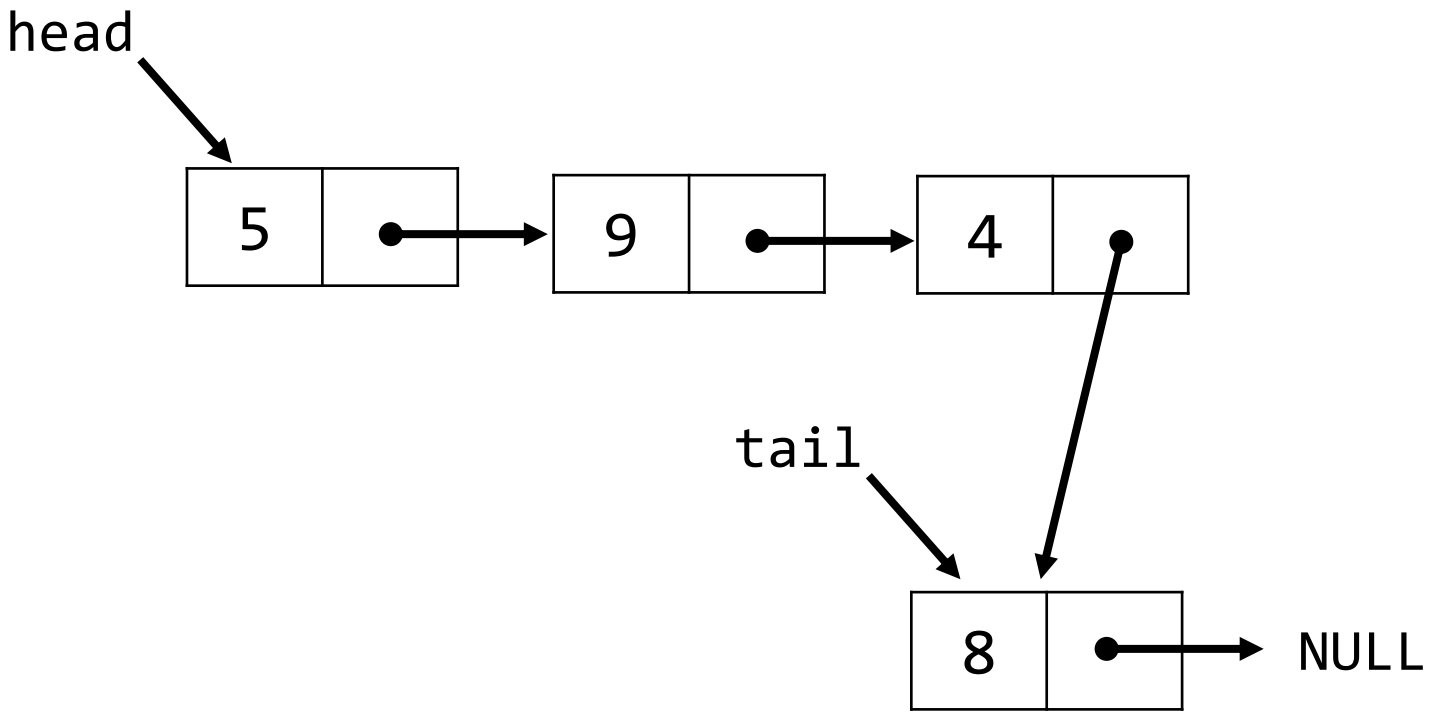
Implementation Using Linked List

Push_back Implementation



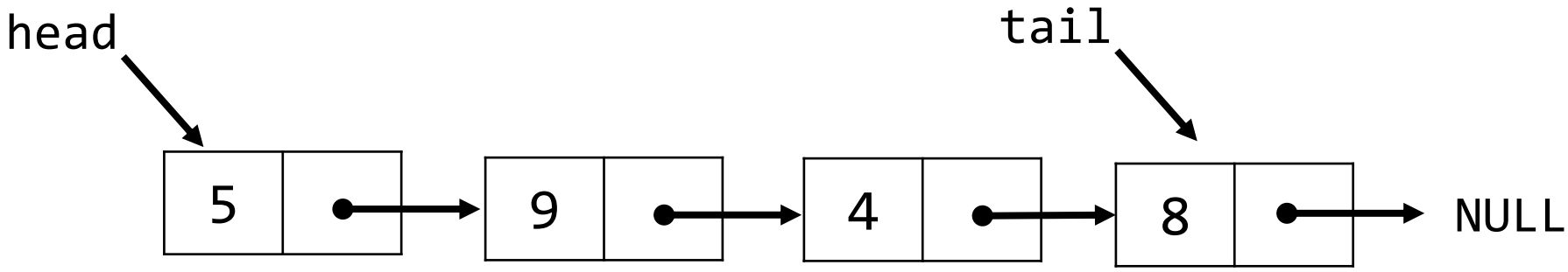
Implementation Using Linked List

Push_back Implementation



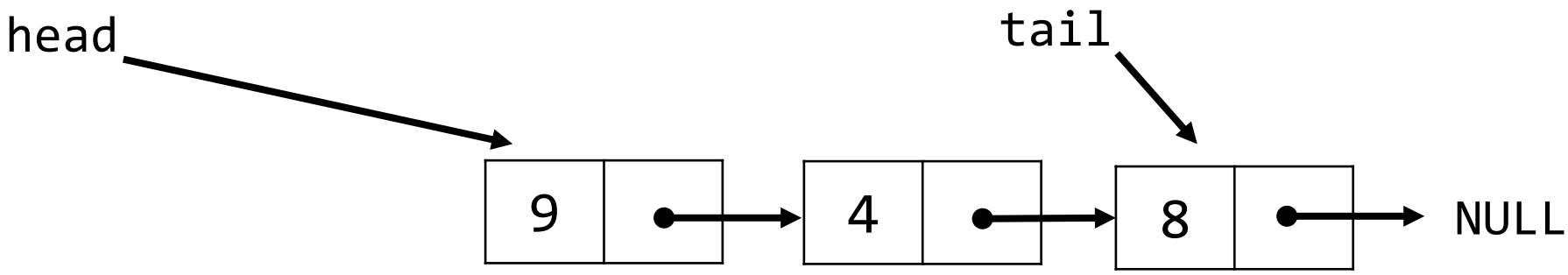
Implementation Using Linked List

How to implement tail?



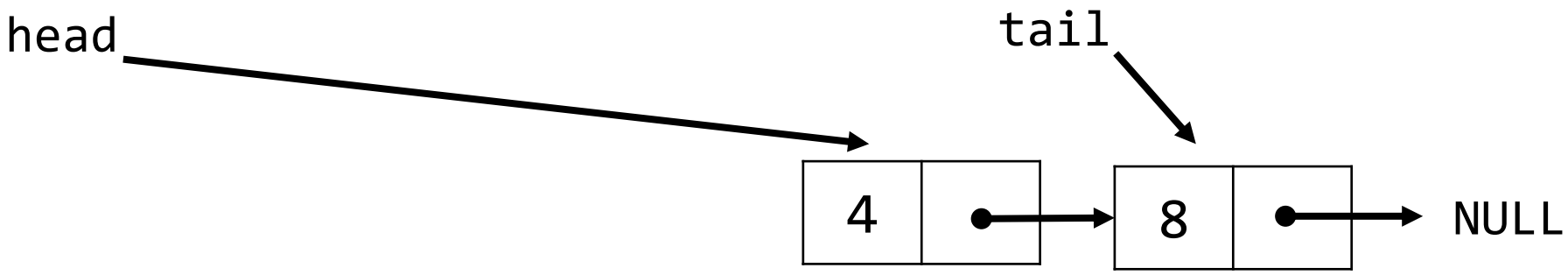
Implementation Using Linked List

Pop_front()



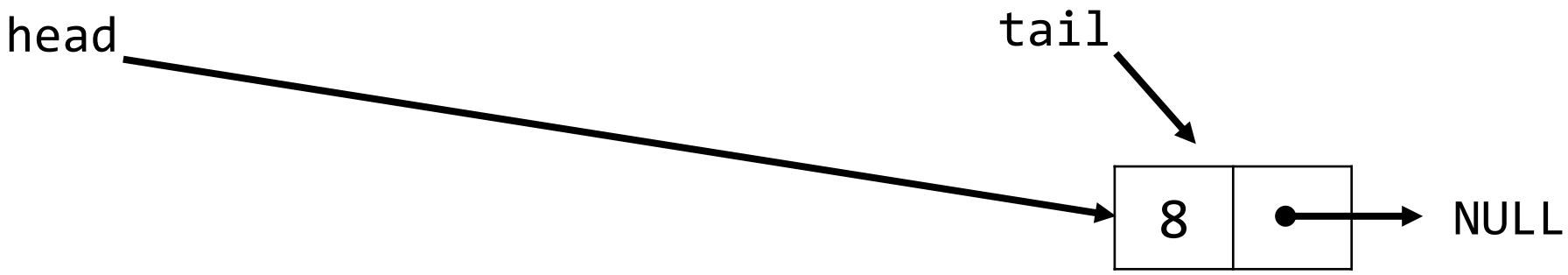
Implementation Using Linked List

Pop_front()



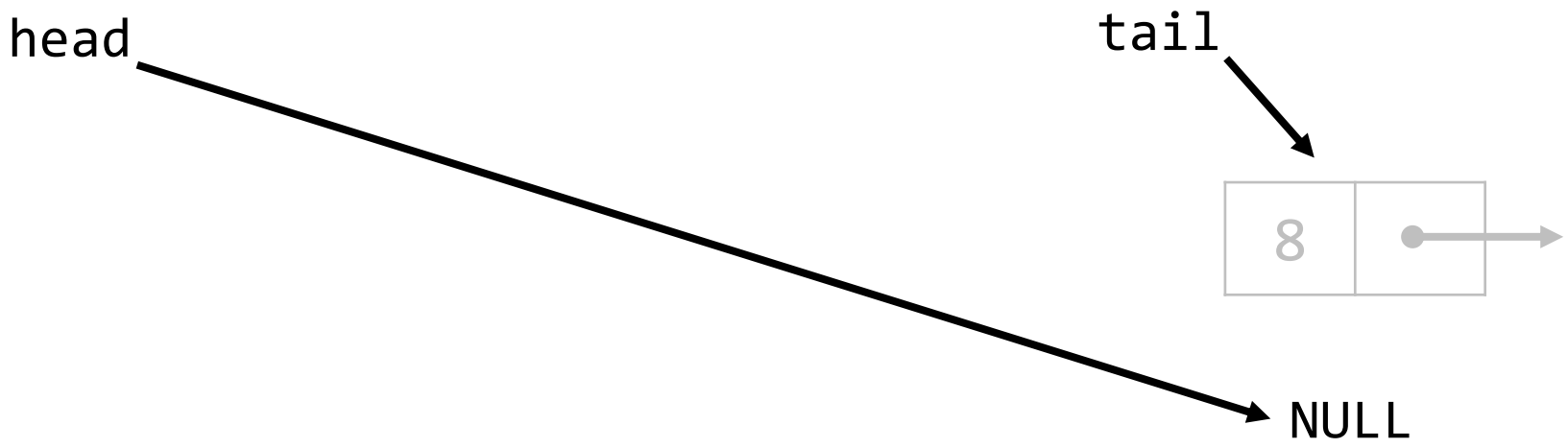
Implementation Using Linked List

Pop_front()



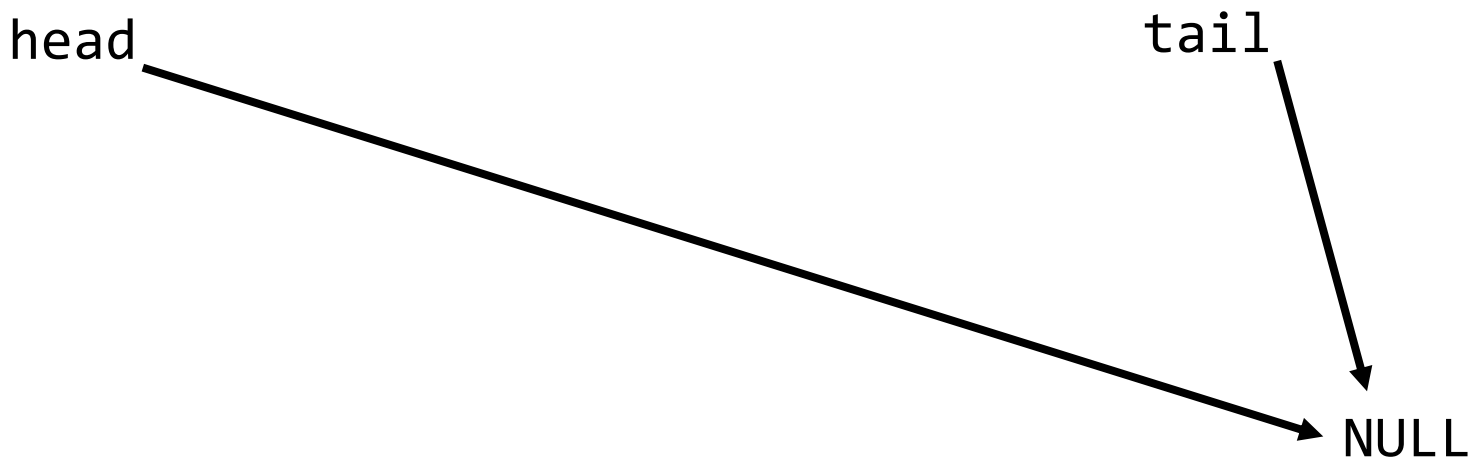
Implementation Using Linked List

Pop_front()
Tail should be updated



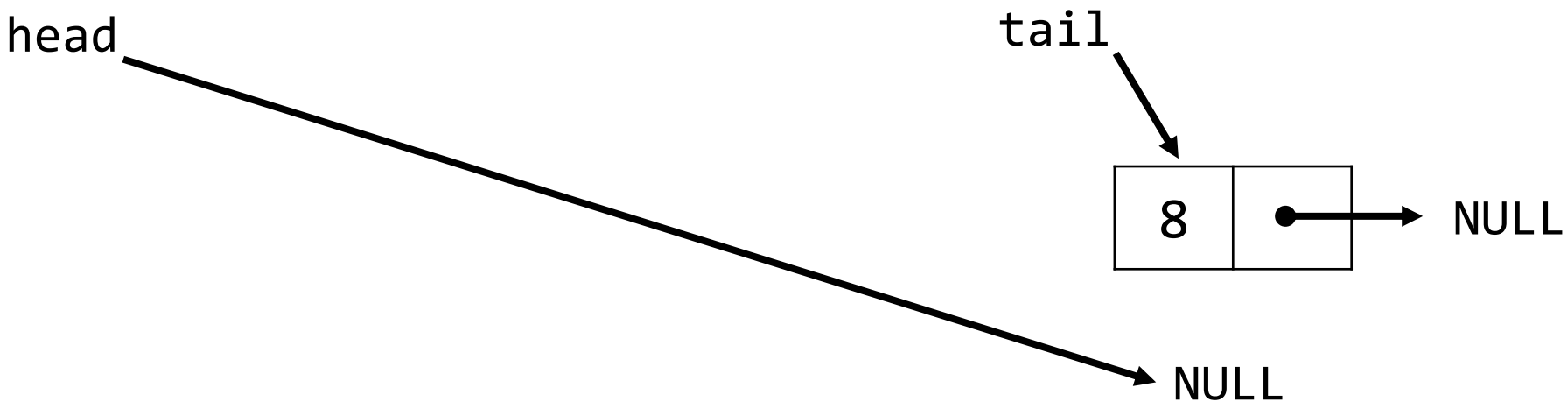
Implementation Using Linked List

Pop_front()
Tail should be updated



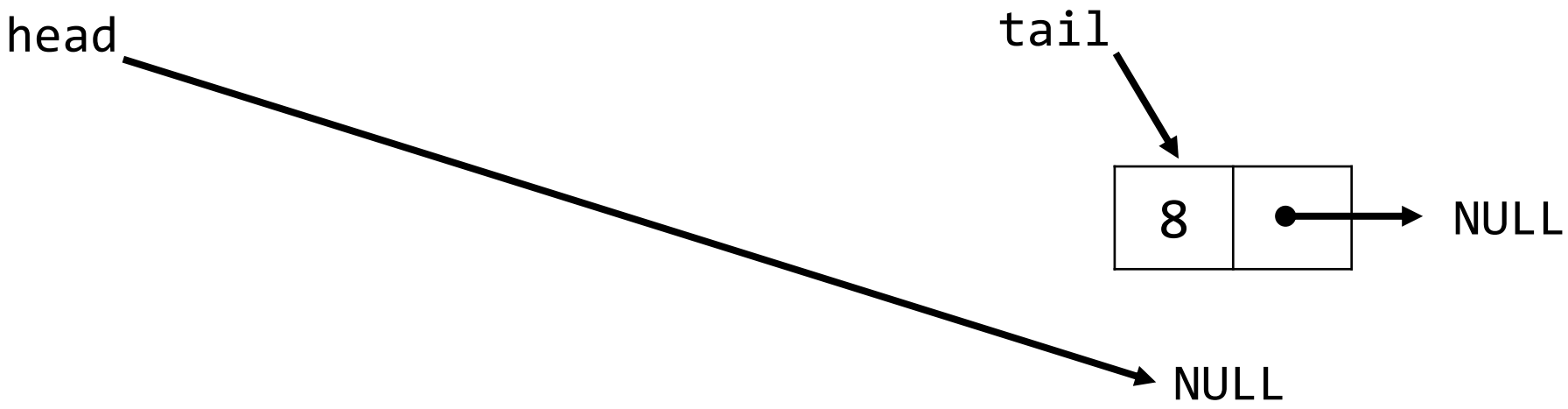
Implementation Using Linked List

Push_back()



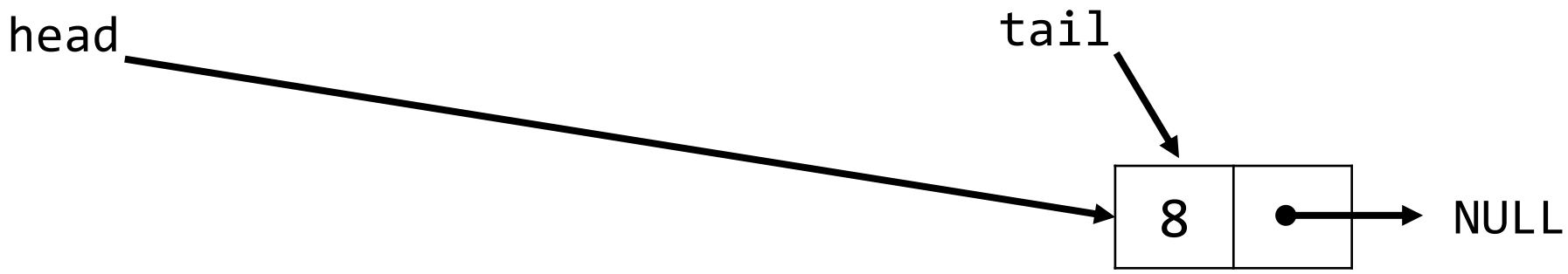
Implementation Using Linked List

Push_back()
Head should be updated



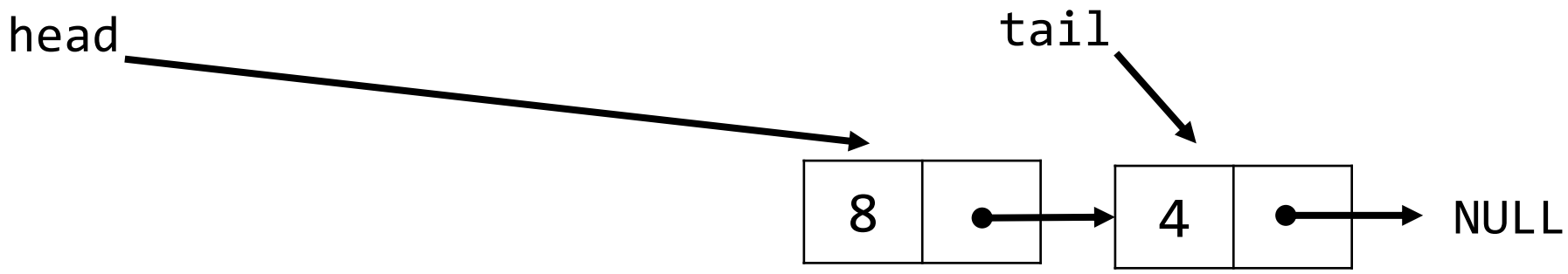
Implementation Using Linked List

Push_back()
Head should be updated



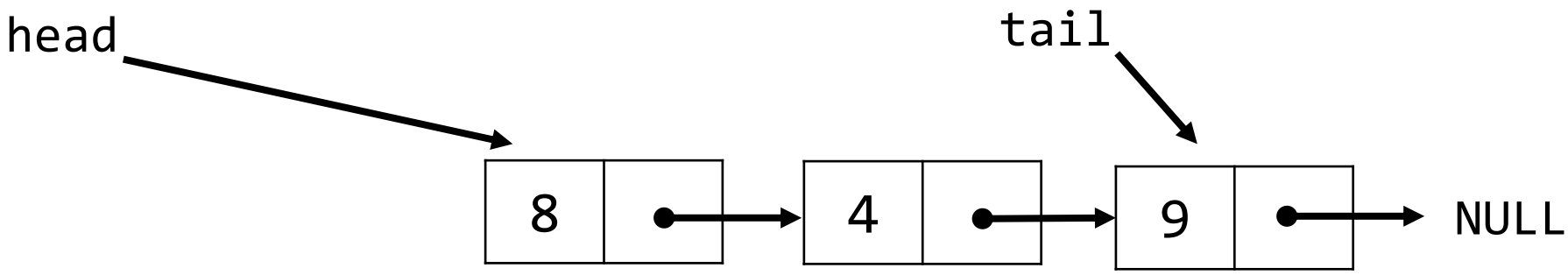
Implementation Using Linked List

Push_back()
Head will not be update after the first insert



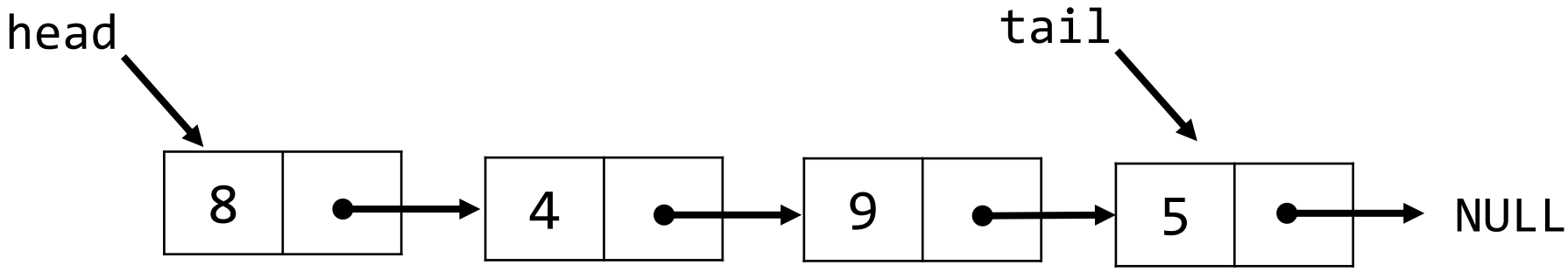
Implementation Using Linked List

Push_back()
Head will not be update after the first insert



Implementation Using Linked List

Push_back()
Head will not be update after the first insert



Array vs Linked List

Which one is preferable in actual performance?

Array vs Linked List

Which one is preferable in **actual performance**?

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Locality of reference

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.

Find sources: "Locality of reference" – news · newspapers · books · scholar · JSTOR *(July 2008) (Learn how and when to remove this template message)*

In computer science, **locality of reference**, also known as the **principle of locality**,[1] is the tendency of a processor to access the same set of memory locations repetitively over a short period of time.[2] There are two basic types of reference locality – temporal and spatial locality. Temporal locality refers to the reuse of specific data, and/or resources, within a relatively small time duration. Spatial locality refers to the use of data elements within relatively close storage locations. Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, such as, traversing the elements in a one-dimensional array.