

Overflow Underflow

Array, Linked List, Stack, Queue

Overflow:

Overflow is a condition where a piece of code is trying to insert a value or create a new memory location which is not possible due to less memory or size.

Array: In array overflow condition occurs when code will try to access the index position beyond the allocated size. In several languages, it returns a garbage value or it will return some error.

Linked List: In the linked list, as it is a dynamic data structure overflow is only possible memory is not available for allocation and there is no free space.

Stack: Stack is a data structure where elements are accessed using a variable named "Top". Overflow happens because of the following condition:

$\text{Top} \geq \text{Size_of_Stack}$

Queue: Queue is a data structure where elements are inserted using "Rear". To identify the overflow condition in queue please check for the following:

$\text{Rear} \geq \text{Size_of_Queue}$

Underflow:

Underflow is a condition where code is trying to access the data even though if there is no data available to access or allocated memory is released.

Array: In array, underflow condition comes up when code tries to access even though there is no element available in the array.

Linked List: Underflow condition occurs in the linked list when allocated memory is already released and there is no element to access or delete.

Stack: In stack, every element gets accessed through "Top". In stack, underflow happens when there is no elements are available to delete or access. To identify the underflow situation please check the following condition.

$\text{if}(\text{Top} \leq -1)$

Queue: In Queue, deletion will happen through "Front". Underflow will happen code is trying to delete or access even though if there is no element in the Queue. In order to check the underflow condition please check for the following.

$\text{Front} == \text{Rear}$