

GeeksQuiz

Computer science mock tests for geeks

- [Home](#)
- [Latest Questions](#)
- [Articles](#)
- [C/C++ Programs](#)
- [Contribute](#)
- [Books](#)

[Subscribe](#)

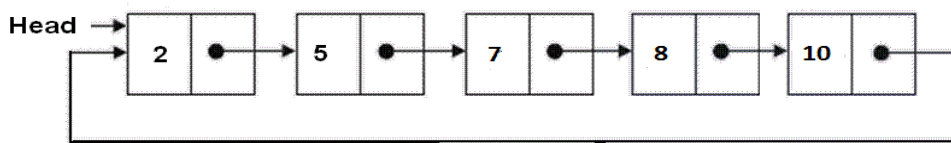
Circular Linked List | Set 1 (Introduction and Applications)

April 4, 2014

We have discussed singly and doubly linked lists in the following posts.

[Introduction to Linked List & Insertion](#)
[Doubly Linked List Introduction and Insertion](#)

***Circular linked list** is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list.*



Advantages of Circular Linked Lists:

- 1) Any node can be a starting point. We can traverse the whole list by starting from any point. We just need to stop when the first visited node is visited again.
- 2) Useful for implementation of queue. Unlike [this](#) implementation, we don't need to maintain two pointers for front and rear if we use circular linked list. We can maintain a pointer to the last inserted node and front can always be obtained as next of last.
- 3) Circular lists are useful in applications to repeatedly go around the list. For example, when multiple applications are running on a PC, it is common for the operating system to put the running applications on a list and then to cycle through them, giving each of them a slice of time to execute, and then making them wait while the CPU is given to another application. It is convenient for the operating system to use a circular list so that when it reaches the end of the list it can cycle around to the front of the list. (Source

<http://web.eecs.utk.edu/~bvz/cs140/notes/Dllists/>)

4) Circular Doubly Linked Lists are used for implementation of advanced data structures like [Fibonacci Heap](#).

We will soon be discussing implementation of insert delete operations for circular linked lists.

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem

Related Questions:

- [Binary Heap](#)
- [Delete all occurrences of a given key in a linked list](#)
- [How to create mergable stack?](#)
- [Deque | Set 1 \(Introduction and Applications\)](#)
- [A data structure for n elements and O\(1\) operations](#)
- [Convert left-right representation of a binary tree to down-right](#)
- [Print level order traversal line by line](#)
- [C Program for Red Black Tree Insertion](#)

Like { 6

Tweet { 1

+1 { 1

9 Comments

GeeksQuiz

Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾



Join the discussion...



typing.. • 10 months ago

we can make circular LL, a more efficient data structure by making another pointer like head, *tail.. This can provide insertion, deletion, in constant time and also make other functions more efficient..

1 ^ | ▾ • Reply • Share ›



Kim Jong-il ➔ typing.. • 6 months ago

@typing.. What is view about point(2). Are not both same, what are you saying and that point.

1 ^ | ▾ • Reply • Share ›



typing.. ➔ Kim Jong-il • 6 months ago

Yes!! both are same.. infact no need to maintain head.. only tail is sufficient..

^ | ▾ • Reply • Share ›

 |  • [Reply](#) • [Share](#) ›**Kartik Nagpal**  typing.. • 6 months ago

In Circular LL, there is no logical head or tail. To visualise, think of a geometric circle in a 2-D plane, head or tail depends on whether you looking clockwise or ant-clockwise.

In CLL, you need only one external pointer to the list todo $O(1)$ insertions/deletions, that external pointer is shown as "head" in this post, or you can have it call "tail", as you wish.

 |  • [Reply](#) • [Share](#) ›**typing..**  Kartik Nagpal • 6 months ago

thanks!!

 |  • [Reply](#) • [Share](#) ›**ishan soni**  typing.. • 6 months ago

why need 2 pointers if we can achieve the goal by maintaining a single pointer (tail)

 |  • [Reply](#) • [Share](#) ›**typing..**  ishan soni • 6 months ago

Yess!! no need of 2 pointers, only tail pointer is sufficient..

 |  • [Reply](#) • [Share](#) ›**Dyaan** • 4 months ago

whats cyclic list?? how to find the given list is cyclic or not??

 |  • [Reply](#) • [Share](#) ›**Kartik Nagpal**  Dyaan • 2 months ago

If you start at any given node in a linked list, and then if by following chain of links/pointers from one node to next node, you can reach the same node you initially started with, thats a cyclic/circular linked list.

Heres the post: <http://www.geeksforgeeks.org/w...>

 |  • [Reply](#) • [Share](#) › [Subscribe](#) [Add Disqus to your site](#) [Privacy](#)



• Categories

- [Articles](#) (131)
 - [Algorithms](#) (24)
 - [C](#) (17)
 - [C++](#) (17)
 - [Data Structures](#) (29)
 - [DBMS](#) (1)
 - [Interview Experiences](#) (6)
 - [Java](#) (2)
 - [Operating Systems](#) (1)
 - [Puzzle](#) (18)
 - [Searching and Sorting](#) (10)
- [Programs](#) (35)
- [Quizzes](#) (2,107)
 - [Aptitude](#) (2)
 - [Computer Science Quizzes](#) (2,104)
 - [Algorithms](#) (147)
 - [C](#) (208)
 - [C++](#) (129)
 - [Computer Organization and Architecture](#) (1)
 - [Data Structures](#) (132)
 - [DBMS](#) (2)
 - [GATE](#) (1,406)
 - [Java](#) (51)
 - [Operating Systems](#) (28)
 - [Web technologies](#) (1)

• Recent Discussions

- [Ajoy](#)

In the dequeue function after dequeuing the...

[Queue | Set 1\(Introduction and Array Implementation\)](#) · [5 hours ago](#)

- [kaps](#)

can we solve the above using 1 pointer only...

[Linked List | Set 3 \(Deleting a node\)](#) · [6 hours ago](#)

- [kaps](#)

refer to previous post

[Linked List | Set 3 \(Deleting a node\)](#) · [6 hours ago](#)

- [Piyush](#)

Well,i am slightly confused at the part where...

[A Programmer's approach of looking at Array vs. Linked List](#) · [7 hours ago](#)

- Sandeep

Hey , can you please explain me about the array...

[C | Pointer Basics | Question 4](#) · [10 hours ago](#)

- [thevagabond85](#)

and for

Input: arr[] = {1, 1, 1, 1, 1, 1, 1}

[Count 1's in a sorted binary array](#) · [15 hours ago](#)

•

Valid [XHTML Strict 1.0](#)

Powered by [WordPress](#) & [MooTools](#) | MiniMoo 1.3.4