Armen Melkumyan • 1st
Technical / Solutions Architect
1yr • Edited •

From .Net C# interview questions

Question:

What are the benefits and drawbacks of using LinkedList<T> compared to List<T>

Answer:

LinkedList<T> and List<T> are both generic collections but have different performance characteristics and use-cases:

Benefits of LinkedList<T>:

1) Dynamic: Each element (node) contains a reference to the next (and/or previous) node. It can be more memory-efficient when frequently adding and removing nodes, especially in the middle of the list.

2) Performance: Better performance for insertions and deletions within the list, as it just involves changing the node links.

Drawbacks of LinkedList<T>:

1) Memory Overhead: Each node in a linked list requires extra memory for its previous/next references.

2) Access Time: Poor performance for indexed access or search. It requires linear time to traverse the list from the start to reach the desired element.

Benefits of List<T>:

1) Indexed Access: Provides fast indexed access to elements. It's an array-backed list, so accessing an element by index is an O(1) operation.

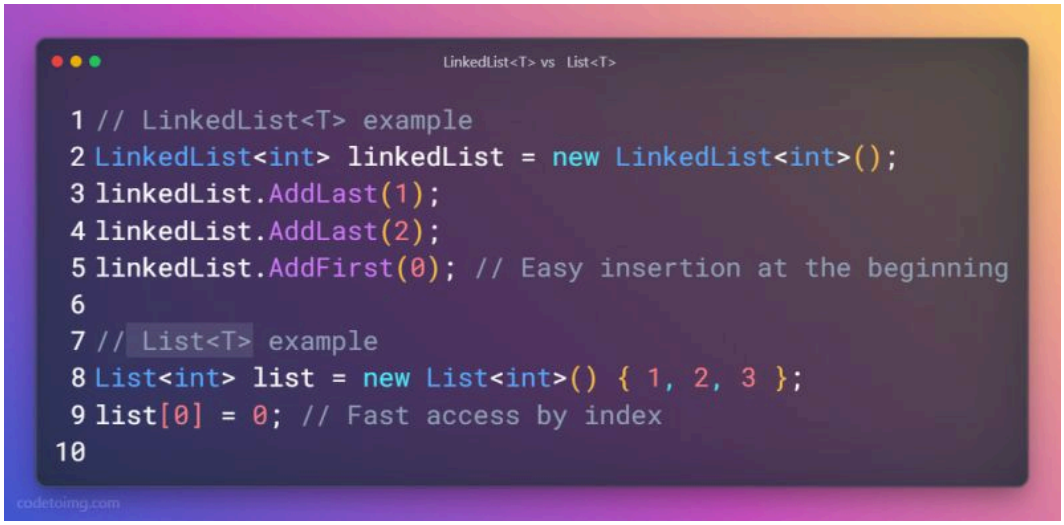
2) Memory: Generally more memory-efficient for a large number of elements as it doesn't require extra space for node references.

Drawbacks of List<T>:

1) Resizing: When the underlying array reaches its capacity, the list needs to be resized, which can be a costly operation.

2) Insertions/Deletions: Inserting or deleting elements in the middle of the list can be costly as it requires shifting elements.

#csharp #dotnet #interview #interviewquestions #interviewpreparation #list #linkedlist



8