Session16 Linked Lists

1. .NET LinkedList class:

* It Implements

System.Collections.Generic.ICollection<T>, System.Collections.Generic.IEnumerable<T>, System.Collections.Generic.IReadOnlyCollection<T>, System.Collections.ICollection,

System.Runtime.Serialization.IDeserializationCallback, System.Runtime.Serialization.ISerializable

* It is a type of Doubly linked list

2. Methods:

* AddAfter
* AddBefore
* AddFirst --same as our implementation
* AddLast --same as our implementation
* Clear—Count is set to zero, First and Last are set to null

Same as our implementation-- Count is set to zero, Head and Tail are set to null

* Contains --same as our implementation
* CopyTo --same as our implementation
* Equals
* Find
* FindLast
* GetEnumerator --same as our implementation
* GetHashCode
* GetObjectData
* GetType
* MemberwiseClone
* OnDeserialization
* Remove --same as our implementation
* RemoveFirst --same as our implementation
* RemoveLast --same as our implementation
* ToString

There are some methods we did not implement in our doubly linked-list such as AddAfter(), AddBefore(), Find() and FindLast()….etc

AddAfter() and AddBefore() provide more flexibilities for adding by specify a particular position to insert new item in a linked list.

In the meanwhile, we implement Add() which calls AddFirst() method.

3. The ability to traverse

In a single linked list, node only points towards next node, and there is no pointer to previous node, which means you cannot traverse back on a singly linked list. On the other hand, doubly linked list maintains two pointers, towards next and previous node, which allows you to navigate in both direction in any linked list.