

1 C#

1.1 Generics

The type constraints for the first version of GreaterCount, it constraints the type T to be implementing the IComparable interface, ensuring that items of type T can be compared between themselves. The generic type U is not having any constraints applied to it, but it is neither used in this case.

Additionally, the item object type needs to be implementing IEnumerable with the generic type T provided.

The type constraints for the second version of GreaterCount, we achieve the same result as the above answer, but through transitivity where the constraints are that T is of same type of U and type U implements the IComparable type with type U. This is only accepted if the IComparable interface allows for implementations for the type U.