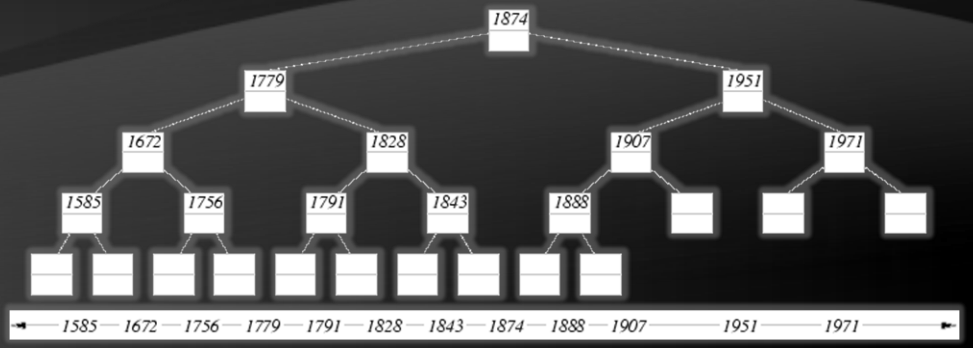
Telerik Academy



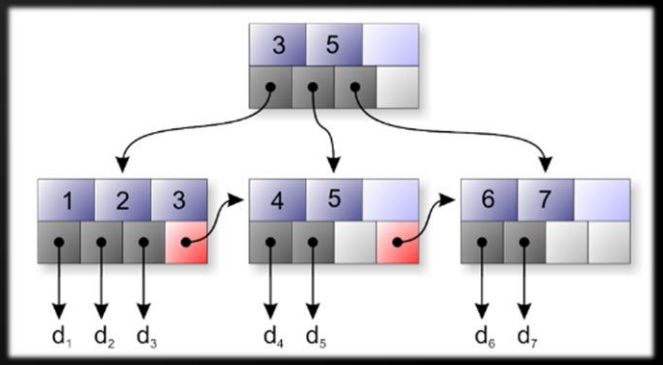
Advanced Data Structures

Wintellect Power Collections, C5 Collections

Data Structures and Algorithms

Telerik Software Academy

http://academy.telerik.com

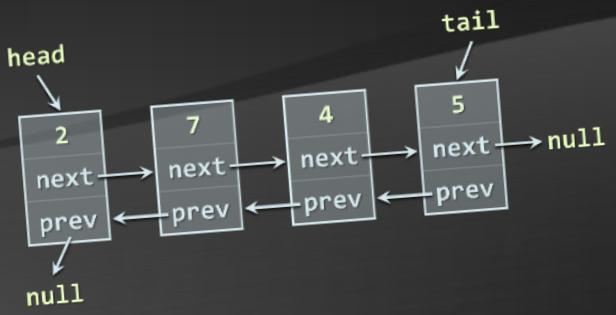


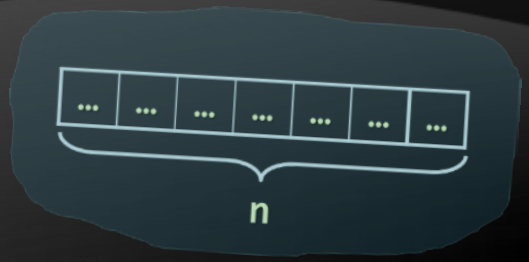
Telerik Academy

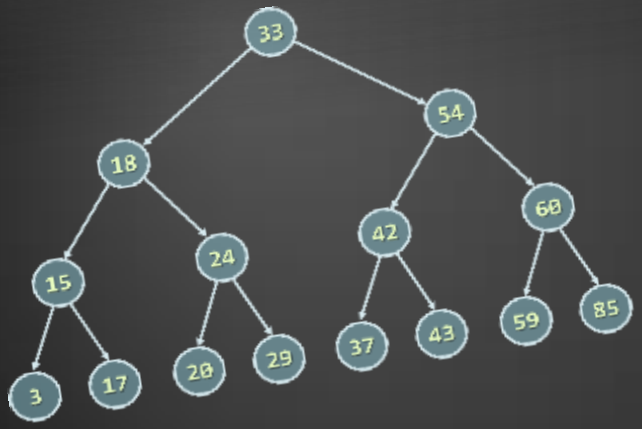
Table of Contents

1. Standard .NET Data Structures
   - Special .NET Collections
2. Wintellect Power Collections
   - Installation
   - Power Collection Classes
   - Implementing Priority Queue
3. C5 Collections
4. Other Advanced Data Structures
   - Suffix trees, interval trees, ropes, tries, etc.

Telerik Academy







Standard .NET Data Structures

Built-In .NET Data Structure Implementations

Telerik Academy

.NET Data Structures

- Linear structures
  - Lists: `List<T>`, `LinkedList<T>`
  - Stacks: `Stack<T>`
  - Queues: `Queue<T>`

- Dictionaries (maps)
  - `Dictionary<K,V>`, `SortedDictionary<K,V>`
  - No standard multi-dictionary .NET class
- Balanced search tree structures
  - `SortedSet<T>`, `SortedDictionary<K,V>`

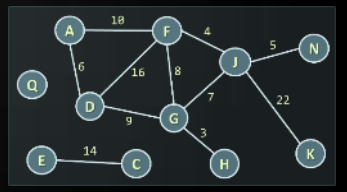
Telerik Academy

.NET Data Structures (2)

- Sets and bags
  - Sets – `HashSet<T>`, `SortedSet<T>`
  - Bag – no standard .NET class

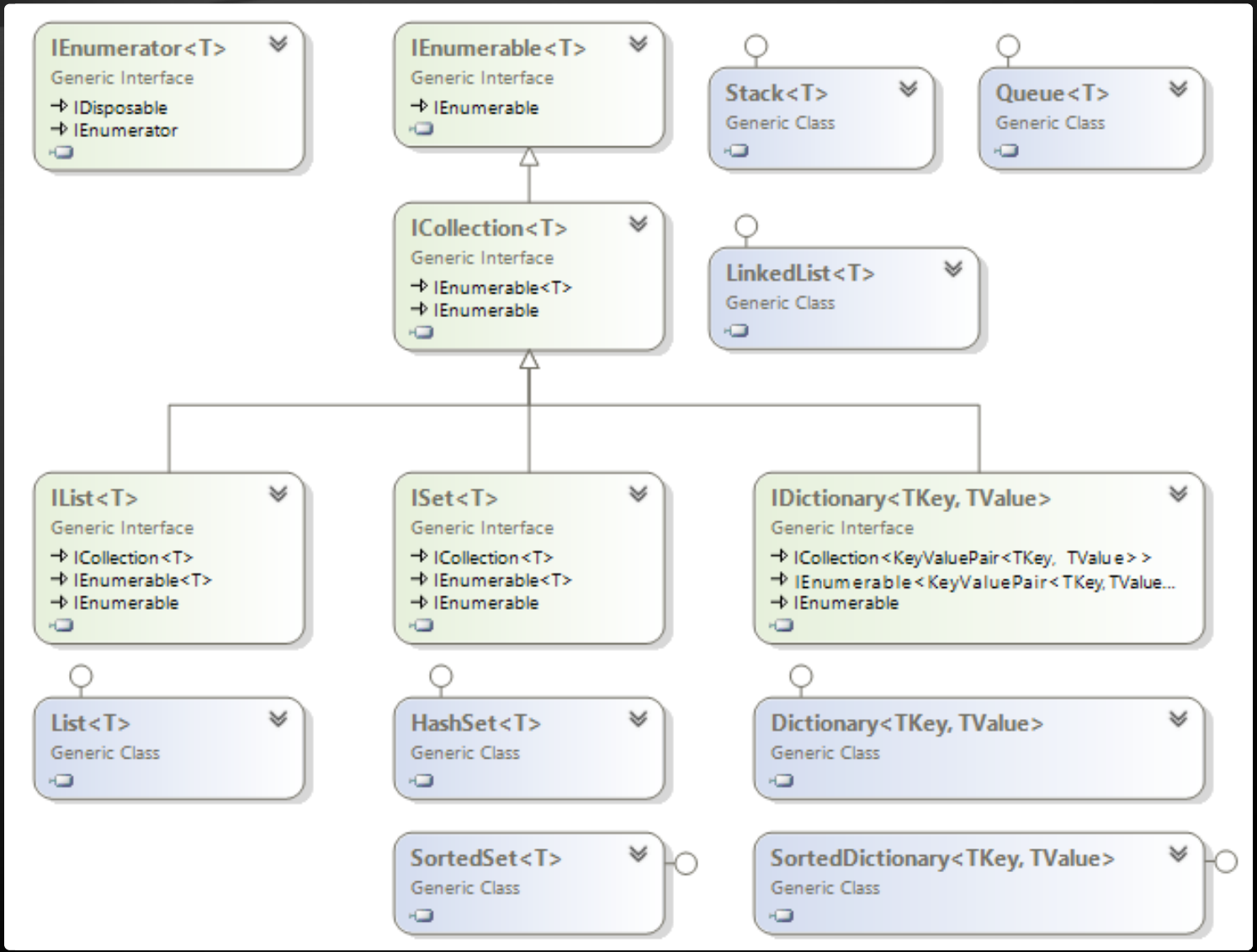
- Ordered sets, bags and dictionaries
- Priority queues / heaps → no
- Special tree structures → no
  - Suffix tree, interval tree, index tree, trie

- Graphs → no
  - Directed / undirected, weighted / un-weighted, connected/ non-connected, …



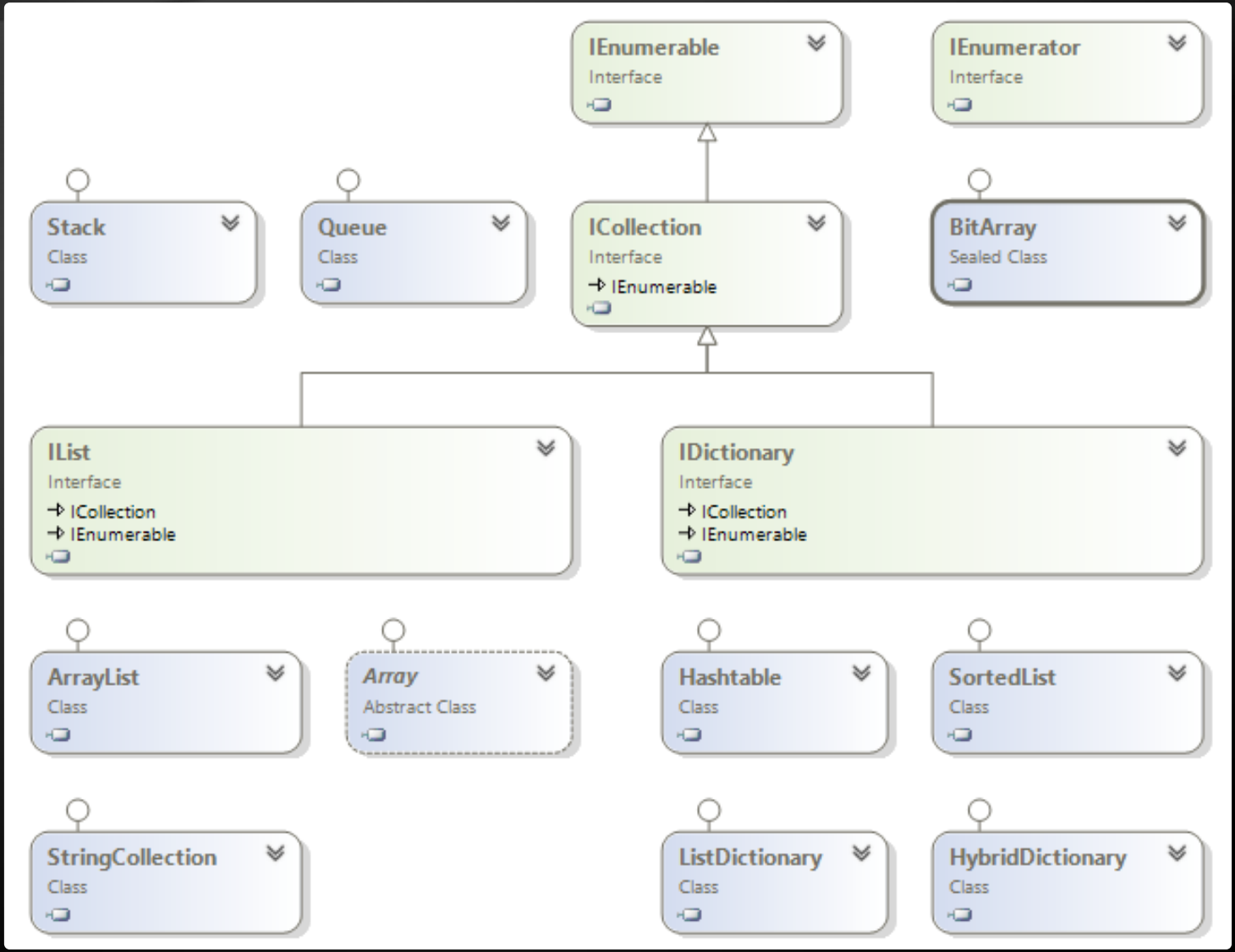
Telerik Academy

.NET Generic Collections



Telerik Academy

.NET Untyped Collections



Special .NET Collections

- Collection<T>
  - Inheritable IList<T>, virtual Add() / Remove()
- ObservableCollection<T>
  - Event CollectionChanged
- IReadOnlyCollection<T>
  - Supports only Count and GetEnumerator()
- IReadOnlyList<T>
  - Supports only Count, [] and GetEnumerator()
- Concurrent collections (thread-safe)
  - BlockingCollection<T>, ConcurrentBag<T>, …

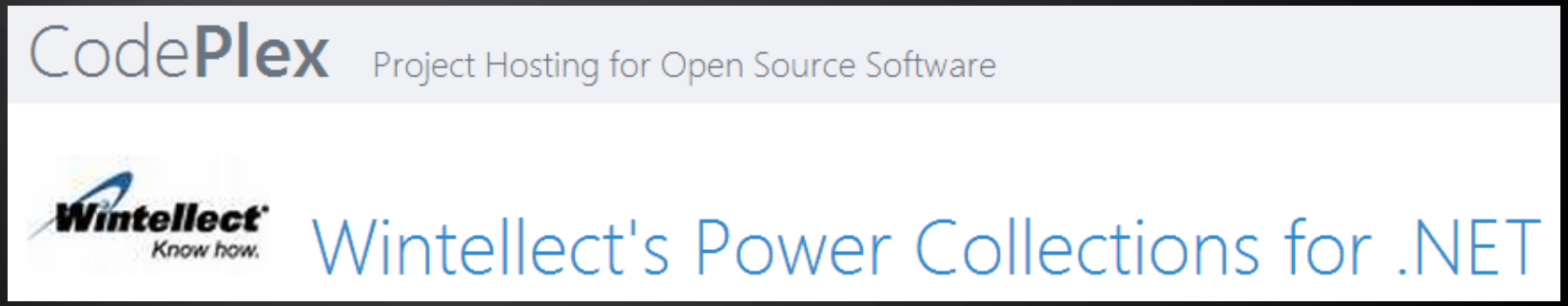
Telerik Academy

Special .NET Collections

Live Demo

Telerik Academy

Wintellect Power Collections

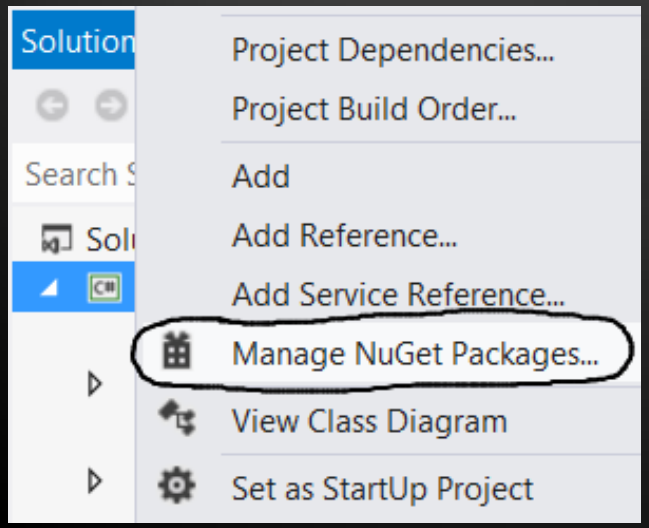


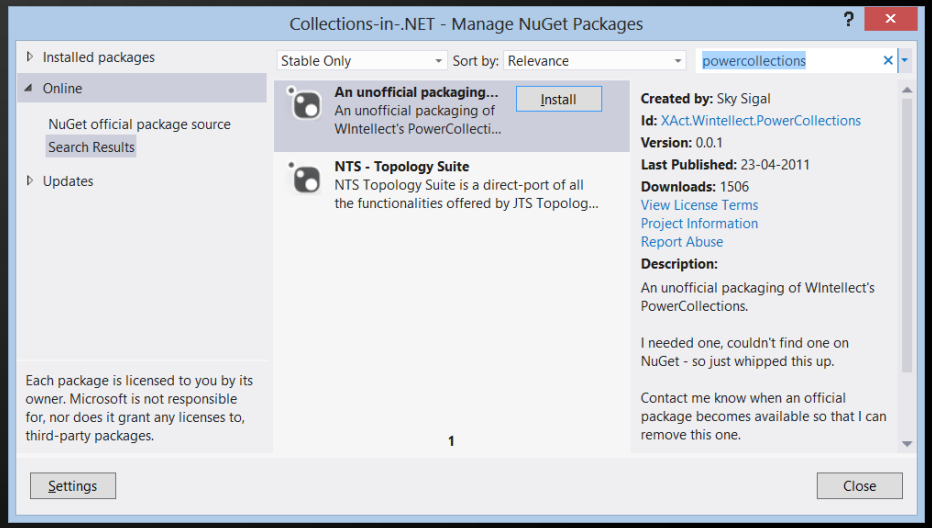
Open Source C# Implementation of All Major Data Structures: Lists, Sets, Bags, Dictionaries, etc.

Telerik Academy

Wintellect Power Collections

- Wintellect Power Collections is powerful open-source data structure library
  - Download: http://powercollections.codeplex.com
- Installing Power Collections in Visual Studio
  - Use NuGet package manager





Telerik Academy

Power Collections Classes

- Bag<T>
  - A bag (multi-set) based on hash-table
    - Unordered collection (with duplicates)
  - Add / Find / Remove work in time O(1)
  - T should provide Equals() and GetHashCode()
- OrderedBag<T>
  - A bag (multi-set) based on balanced search tree
  - Add / Find / Remove work in time O(log(N))
  - T should implement IComparable<T>

Power Collections Classes (2)

- `Set<T>`
  - A set based on hash-table (no duplicates)
  - Add / Find / Remove work in time O(1)
  - Like .NET's `HashSet<T>`
- `OrderedSet<T>`
  - A set based on balanced search tree (red-black)
  - Add / Find / Remove work in time O(`log(N)`)
  - Like .NET's `SortedSet<T>`
  - Provides fast `.Range(from, to)` operation

Telerik Academy

Power Collections Classes (3)

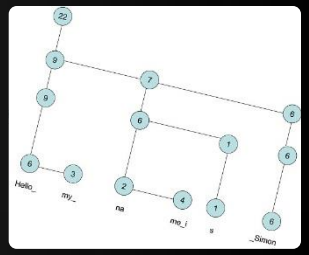
- `MultiDictionary<TKey, TValue>`
  - A dictionary (map) implemented by hash-table
  - Allows duplicates (configurable)
  - Add / Find / Remove work in time O(1)
  - Like `Dictionary<TKey, List<TValue>>`
- `OrderedDictionary<TKey, TValue>` / `OrderedMultiDictionary<TKey, TValue>`
  - A dictionary based on balanced search tree
  - Add / Find / Remove work in time O(log(N))
  - Provides fast `.Range(from, to)` operation

Telerik Academy

Power Collections Classes (4)

- Deque<T>
  - Double-ended queue (deque)

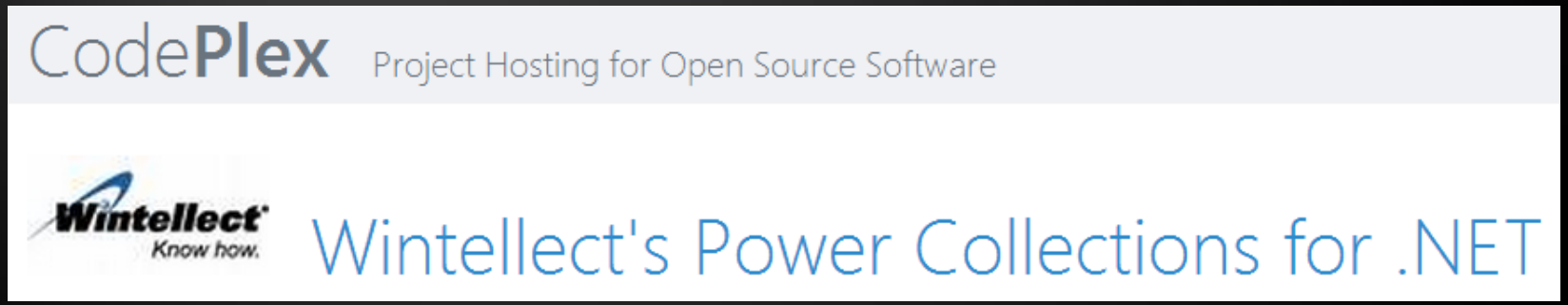
- BigList<T>
  - Editable sequence of indexed items
  - Like List<T> but provides
    - Fast Insert / Delete operations (at any position)
    - Fast Copy / Concat / Sub-range operations
  - Implemented by the data structure "Rope"
    - Special kind of balanced binary tree: http://en.wikipedia.org/wiki/Rope_(data_structure)



Telerik Academy

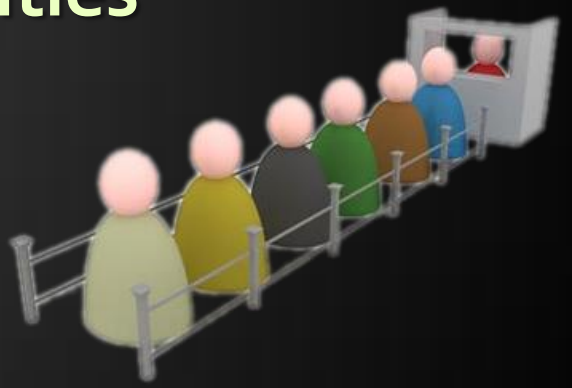
Wintellect Power Collections

Live Demo



Priority Queue

- What is a "priority queue"?
  - Data structure to efficiently support finding the item with the highest priority
  - Like a queue, but with priorities
  - The basic operations
    - `Enqueue(T element)`
    - `Dequeue() → T`



- There is no build-in priority queue in .NET
  - See the data structure "binary heap"
  - Can be implemented also by `OrderedBag<T>`

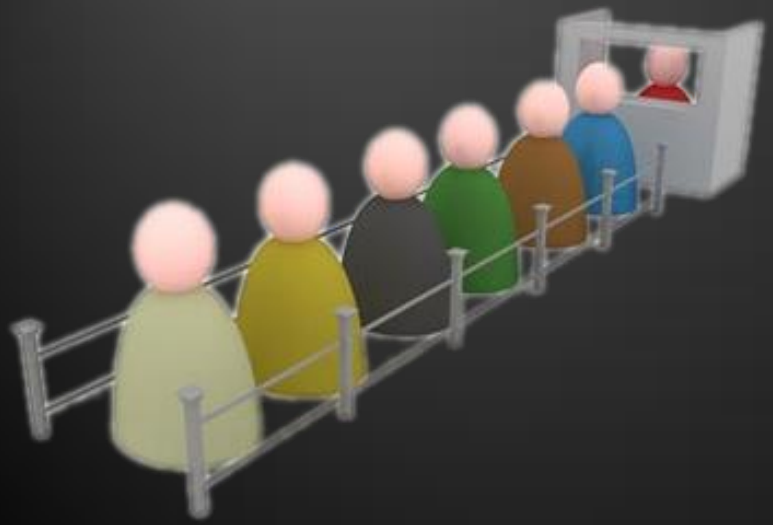
Priority Queue Implementation

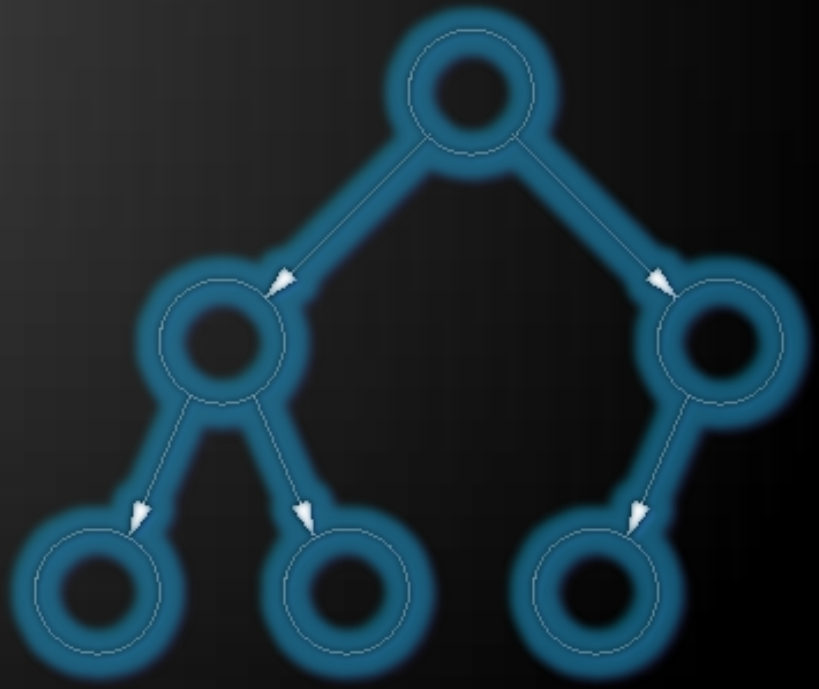
```
class PriorityQueue<T> where T : IComparable<T>
{
    private OrderedBag<T> queue;
    public int Count
    {
        get { return this.queue.Count; }
    }
    public PriorityQueue()
    {
        this.queue = new OrderedBag<T>();
    }
    public void Enqueue(T element)
    {
        this.queue.Add(element);
    }
    public T Dequeue()
    {
        return this.queue.RemoveFirst();
    }
}
```

Telerik Academy

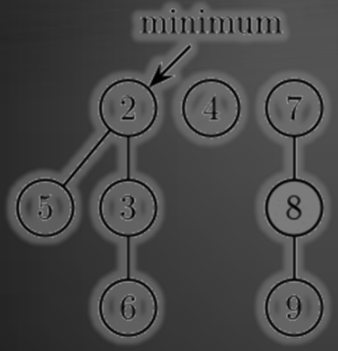
Priority Queue

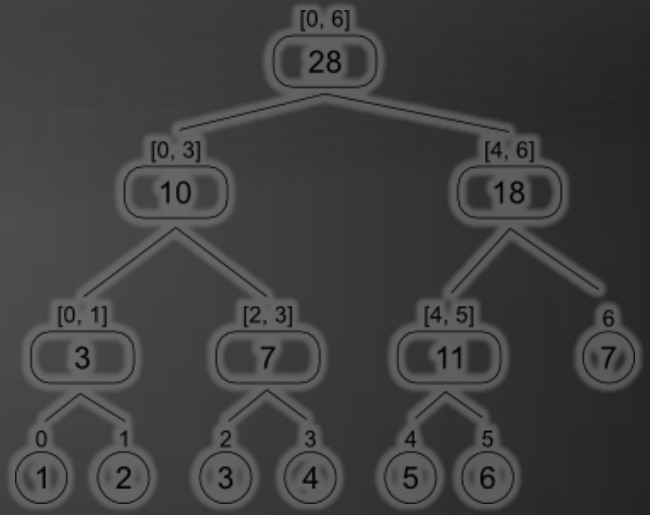
Live Demo

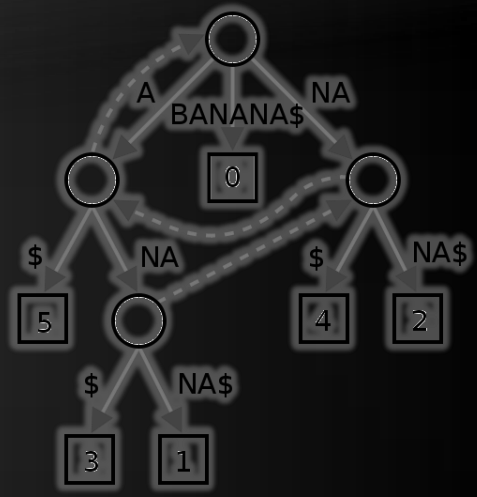




Telerik Academy





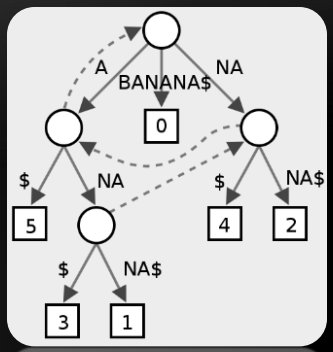


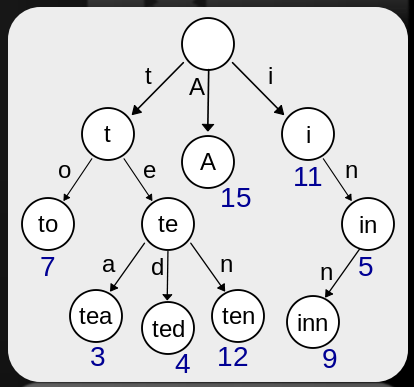
Advanced Data Structures

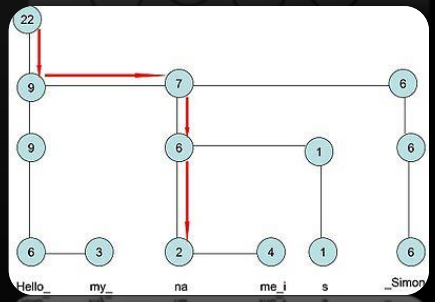
Suffix Trees, Interval Trees, Tries, Ropes, Heaps, …

Advanced Data Structures

- Suffix tree (position tree)
  - Represents the suffixes of given string
  - Used to implement fast search in string
- Trie (prefix tree)
  - Special tree structure used for fast multi-pattern matching
- Rope
  - Balanced tree structure for indexed items with fast inserts / delete
  - Allows fast string edit operations

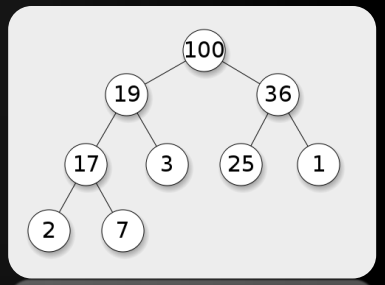






Advanced Data Structures (2)

- Interval tree
  - Keeps intervals [a…b] in ordered balanced tree
  - Allows to efficiently find all intervals that overlap with any given interval or point
- Binary heap, Fibonacci heap
  - Special tree-like data structures to efficiently implement a priority queue

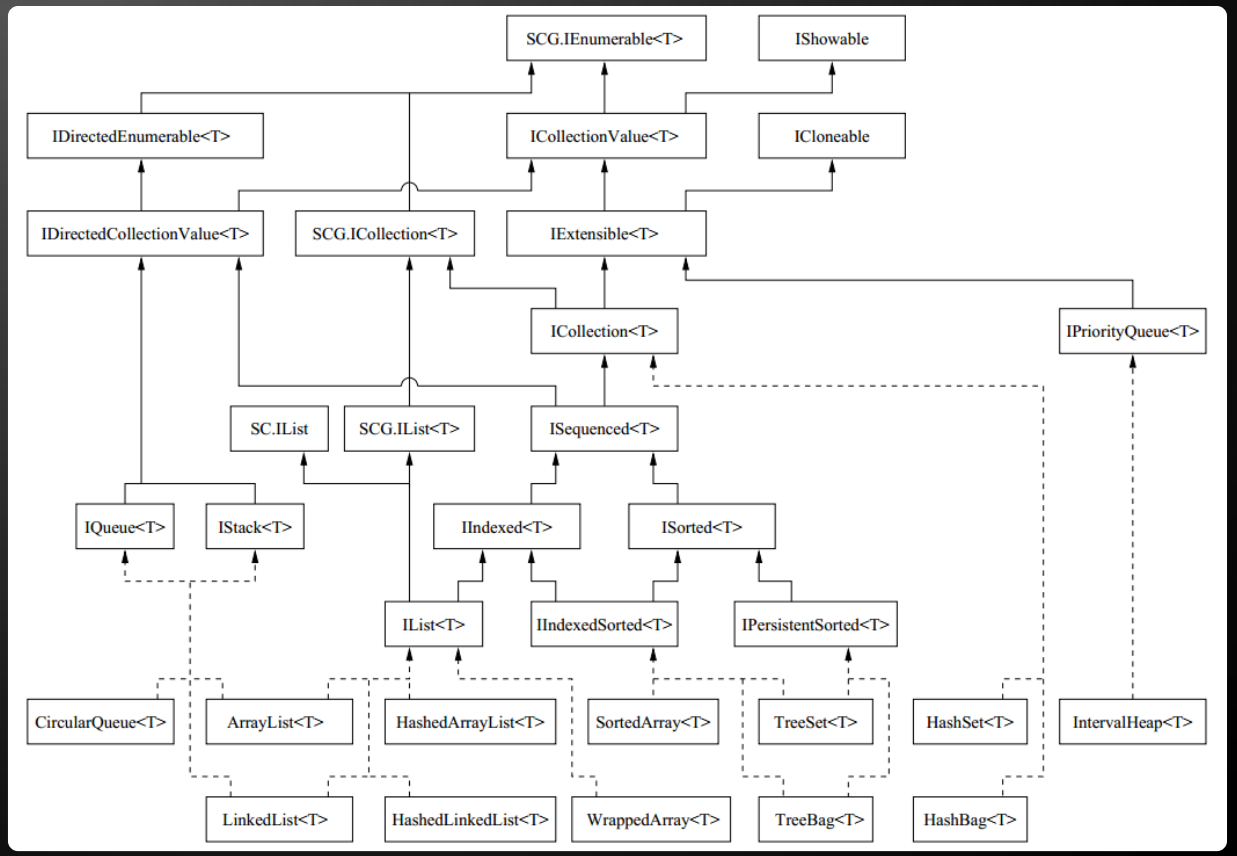


- Index trees
  - Used to keep sorted indices of database records
  - B-tree, B+ tree, T-tree

Telerik Academy

C5 Collections

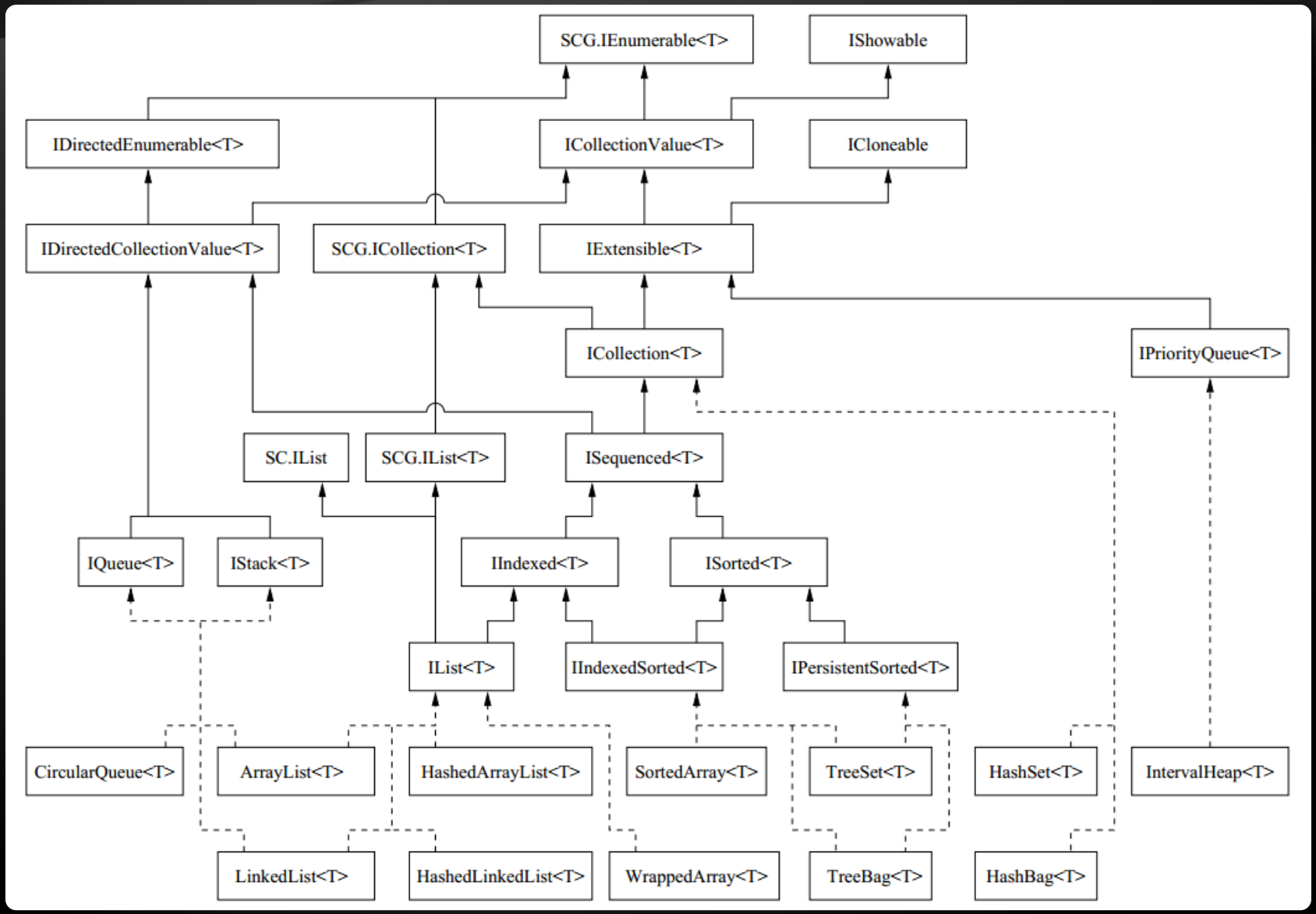
Open Source Generic Collection Library for C#



C5 Collections

- What are "C5 Collections"?
  - C5 Generic Collection Library for C# and CLI
  - Open-Source Data Structures Library for .NET
  - http://www.itu.dk/research/c5/
  - Have solid documentation (book) – http://www.itu.dk/research/c5/latest/ITU-TR-2006-76.pdf
  - The C5 library defines its own interfaces like `IEnumerable<T>`, `IIndexed<T>`, `IIndexedSorted<T>`, etc.

C5 Collection Classes

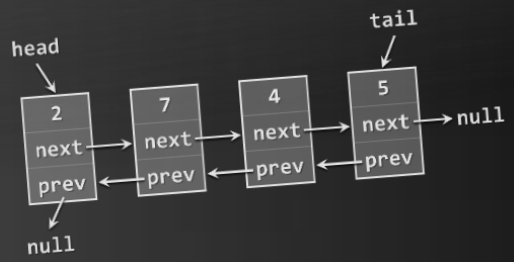


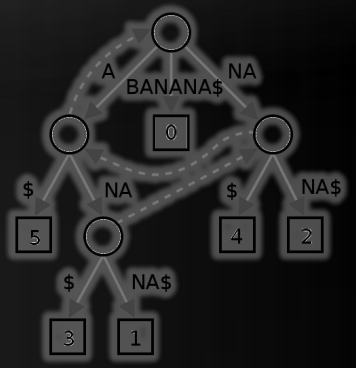
C5 Collection Classes

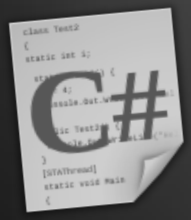
- Classical collection classes
  - `ArrayList<T>`, `LinkedList<T>`, `CircularQueue<T>`, `HashSet<T>`, `TreeSet<T>`, `HashBag<T>`, `TreeBag<T>`
- `HashedArrayList<T>`
  - Combination of indexed list + hash-table
  - Fast Add / Find / indexer [ ] → O(1)
- `IntervalHeap<T>`
  - Efficient double-ended priority queue

Telerik Academy

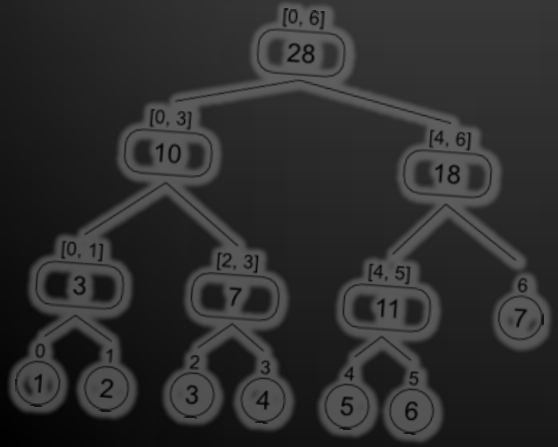
Advanced Data Structures

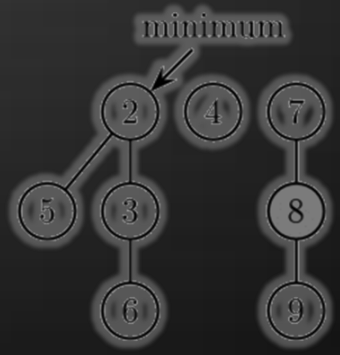






Questions?





http://academy.telerik.com

Exercises

1. Implement a class `PriorityQueue<T>` based on the data structure "binary heap".
2. Write a program to read a large collection of products (name + price) and efficiently find the first 20 products in the price range [a…b]. Test for 500 000 products and 10 000 price searches.

   Hint: you may use `OrderedBag<T>` and sub-ranges.
3. Write a program that finds a set of words (e.g. 1000 words) in a large text (e.g. 100 MB text file). Print how many times each word occurs in the text.

   Hint: you may find a C# trie in Internet.

Telerik Academy

Free Trainings @ Telerik Academy

- C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com