

Sample 1 Readme

This sample is written in pure C# on .NET core. The content concerns high-level data structures and related algorithms. Due to brevity constriction, this code sample contains only several files and therefore cannot be demonstrated or built.

Code Contents

- **IPriorityQueue.cs** Includes an interface `IPriorityQueue`, which indicates that implementing class can be used as a priority queue. There are also `IBinaryTree` and `IHeap` interfaces in the whole project.
- **FibonacciNode.cs** & **FibonacciHeap.cs** These two files describe an implementation of a fibonacci heap, which implements the basic functions of such a heap: `Insert`, `Union`, `ExtractMin`, and `DecreaseKey`.
- **RedBlackTree.cs** This file together with file **RedBlackNode.cs** (curtailed to cut down the codes) describes an implementation of a red-black tree, which implements operations: `Insert`, `Delete`, `Search`, and `Traverse`. As a basic type of balanced tree, this implementation has a search operation with the complexity $O(\log n)$ no matter how "ill" the input data is.
- **Treap.cs** Treap is a data structure can be used as a tree as well a heap. The operations might not be difficult to implement, and the only problem is to implement both tree and heap interfaces.

You may also clone the whole project from [github](#).