K-ичное дерево - дерево, в котором количество ребер, выходящих из вершин, не превышает К (дочерних вершин не более К). При этом троичное дерево является одновременно и четвертичным-просто четвертого ребра нет пока что. В принципе, вершины могут неупорядочены, но нам интересен случай, когда они упорядочены.

Двоичное дерево - упорядоченная несбалансированная (куча - пример абсолютно сбалансированной) система данных.

Как эту штуку хранить? Флешбеки из Вьетнама - односвязный список. Реализуем это так - сделаем два указателя ( на левую вершину и на правую) и указатель на родительскую вершину (parent , left , right). Вот по таким звеньям и будем бегать.

Реализация - в приложенном файле.

Двоичное дерево поиска:

Корневое двоичное упорядоченное дерево, построенное по следующему правилу для любого звена - все ключи левого поддерева key i< key node

правого поддерева key j> key node

В чем удобство? Быстрый поиск. принцип - от бинарного поиска, так как каждый раз отбрасываем половину дерево. Чем оно лучше бинарного поиска в отсортированном массиве? Сложностей будет даже больше, чем кажется сейчас. Причина - скорость поиска и добавления. O(log2(N)). Такая же асимптотика у добавления в дерево. В упорядоченном дереве - добавление элемента за N. А вот что делать в случае удаления? Нужно же перестроить дерево. Тут решение простое. Идем влево от удаляемого элемента и берем самы й правый из них. Ведь для него будет выполняться условие - все элементы, правее исходного, будут больше взятого нами, а все левые - меньшие.