

COS 226 – Data Structures and Algorithms
Fall 2014 – Flipped Lecture Section
Group Worksheet week 5 – 10.09.14
30 minutes

1. Given n random items, prove that it is not possible to build a BST in linear time.

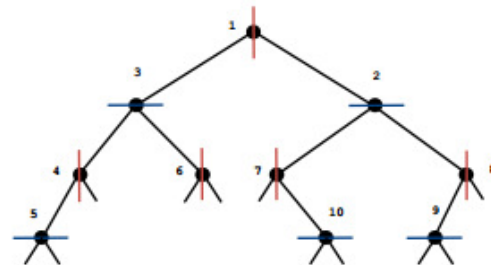
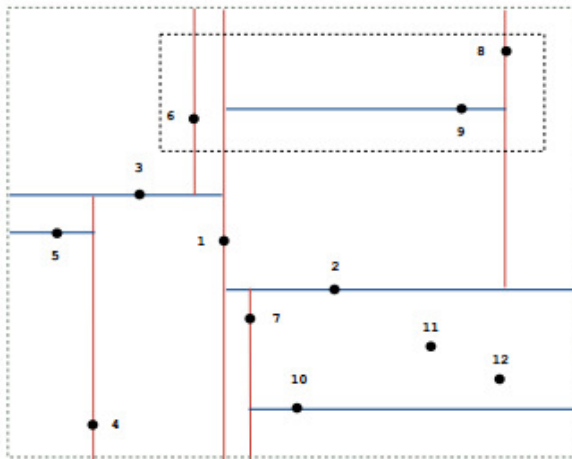
Proof by Contradiction

Suppose we can build a BST in linear time using random elements.
 Now we can use inorder traversal (linear time) to go over the elements and so that we can produce a sorted list.

This means, now we have discovered a comparison based sorting algorithm that is linear time.
 Contradicts our earlier finding that, there exists no comparison based sorting algorithm that is better than $N \lg N$

So our assumption is incorrect and hence we cannot build a BST in linear time.

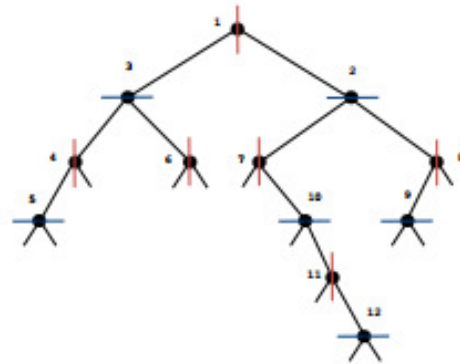
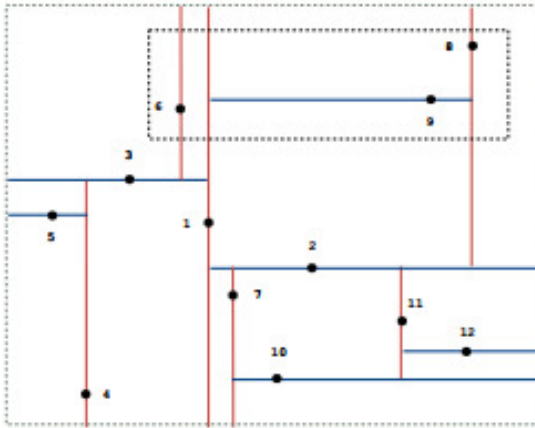
2. The figure below illustrates the results of inserting points 1-10 into a KdTree



- a. What are all the points examined in a query for the points inside the rectangle shown above (in dotted lines)

1 2 3 6 8 9 (though the search may go one extra level, depending on implementation)

b. Draw the result of inserting point 11, then point 12 in the two figures above



3. String symbol table implementation

For each of the operations on the left, list which one or more of the symbol table implementations on the right can be used to *efficiently* implement it. By efficient, we mean $L \log N$ or better on typical ASCII strings (in random order) of average length L , where N is the number of keys in the data structure.

Solution

- | | |
|--|----------------------------------|
| <i>BCDE</i> Find the value associated with a given string key in the data structure. | A. Unordered array. |
| <i>CDE</i> Associate a value with a string key. | B. Ordered array. |
| <i>CDE</i> Delete a string key (and its associated value) from the data structure. | C. Red-black BST. |
| <i>BCE</i> Find the smallest string key in the data structure. | D. Separate-chaining hash table. |
| <i>BCE</i> Find the smallest string key in the data structure that is greater than or equal to a given string. | E. Ternary search trie. |
| <i>E</i> Find the string key in the data structure that is the longest prefix of a given string. | |
| <i>E</i> How many string keys in the data structure starts with a given prefix? | |

Can also be done with an ordered array (B) or a red-black BST (C) by calling `rank()` twice, once with the prefix and once with the last character in the prefix incremented by one.