Data Structures

Binary Search Trees

CS 225
Brad Solomon

July 14, 2022
Contacting CS 225 Admin Staff

Need an extension? Missed an exam? Have a logistic issue or emergency?

Email: cs225admin@lists.cs.illinois.edu
# Exam Logistics

[https://courses.engr.illinois.edu/cs225/fa2022/exams/](https://courses.engr.illinois.edu/cs225/fa2022/exams/)

## Future Exams

<table>
<thead>
<tr>
<th>Exam</th>
<th>Exam 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Exam Availability 09/11 - 09/13</td>
</tr>
</tbody>
</table>

There will be one programming question and a few short answer questions on C++. The programming question will be in an environment.

## Current Exam

<table>
<thead>
<tr>
<th>Exam</th>
<th>60 points</th>
</tr>
</thead>
</table>

## Past Exams

<table>
<thead>
<tr>
<th>Exam</th>
<th>Exam 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Exam Availability 08/30 - 09/01</td>
</tr>
</tbody>
</table>

Exam 0 is designed to be a low-stress introduction to the CBTF exam environment. This quiz is on foundational knowledge you...
Learning Objectives

Conceptualize and code tree traversals

Review dictionaries and binary trees

Introduce the binary search tree

Conceptualize and code BST functions
Traversals
template<class T>
void BinaryTree<T>::__Order(TreeNode * root)
{
"
Traversals

```cpp
template<class T>
void BinaryTree<T>::__Order(TreeNode * root) {
    if (root != NULL) {
        ______________;  
        __Order(root->left);  
        ______________;
        __Order(root->right);  
        ______________;
    }
}
```

```plaintext
+   
|   |
-   *
|   |
  a  /  d  e
  |  |  |
  b  c  
```
Traversals

```
template<class T>
void BinaryTree<T>::__Order(TreeNode * root)
{
    if (root != NULL) {
        ______________;
        __Order(root->left);
        ______________;
        __Order(root->right);
        ______________;
    }
}
```
A Different Type of Traversal
A Different Type of Traversal

template<class T>
void BinaryTree<T>::lOrder(TreeNode * root)
{
    Queue<TreeNode*> q;
    q.enqueue(root);

    while( q.empty() == False){
        TreeNode* temp = q.head();
        process(temp);
        q.dequeue();
        q.enqueue(temp->left);
        q.enqueue(temp->right);
    }
}
Traversals vs Search

Traversals

Search
Dictionary ADT

Data is often organized into key/value pairs:

- Word ➔ Definition
- Course Number ➔ Lecture/Lab Schedule
- Node ➔ Edges
- Flight Number ➔ Arrival Information
- URL ➔ HTML Page
#pragma once

class Dictionary {
    public:

    // ...

    private:

    // ...

};
Binary Trees
Search: Breadth First vs Depth First

Breadth First Search (BFS)

Depth First Search (DFS)
Choosing a search algorithm

The average ‘branch factor’ for a game of chess is ~31. If you were searching a decision tree for chess, which search algorithm would you use?
Improved search on a binary tree

Improved search on a binary tree

7 3 1 5 4 6

1 3 4 5 6 7
Binary Search Tree (BST)

A **binary search tree** $T$ is either:

- _OR_

- _OR_

- _OR_

- _OR_
#pragma once

template <typename K, typename V>
class BST {
public:
    BST();
    void insert(const & K key, V value);
    V remove(const K & key);
    V find(const K & key) const;

private:
    struct TreeNode {
        TreeNode *left, *right;
        K key;
        V value;
        TreeNode();
    };

    TreeNode *head_;
BST Find

find(66)
BST Find

find(9)
template<typename K, typename V>

__________________________ _find(TreeNode *& root, const K & key) {
}
template<typename K, typename V>
V find(const K & key) const {
}
BST Insert

insert(33)
template<typename K, typename V>
void insert(const K & key, V val) {
BST Insert

What binary tree would be formed by inserting the following sequence of integers: [3, 7, 2, 1, 4, 8, 0]
BST Remove

remove(40)
BST Remove

remove(25)
BST Remove

remove(13)
BST Remove

remove(51)
<table>
<thead>
<tr>
<th>Operation</th>
<th>BST Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>find</td>
<td></td>
</tr>
<tr>
<td>insert</td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
</tr>
<tr>
<td>traverse</td>
<td></td>
</tr>
</tbody>
</table>