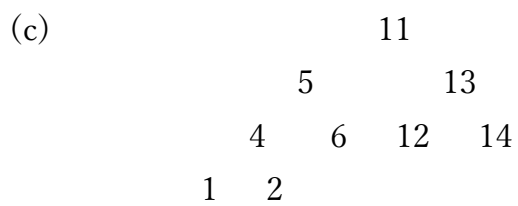
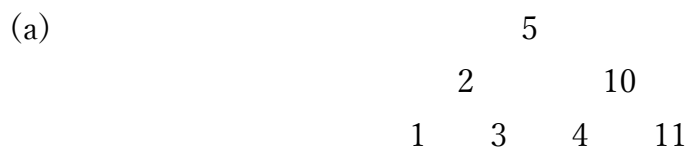


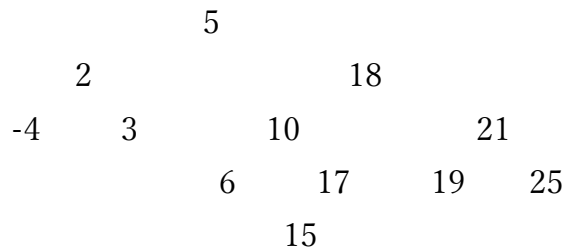
Binary Search Tree

1. Insert 1, 4, 60, -5, 3, 5, 0, 20 into a tree

2. Which of the following Tree is correct?



3. Draw out the path when trying to find “14” in the following Tree



4. The average time required for a search operation in binary tree is

- (a) $O(n)$
- (b) $O(\log_2 n)$
- (c) $O(2n)$
- (d) $O(1)$

5. If the binary tree is not balanced, which is the slowest speed when we search an element inside that tree

- a. $O(\log_2(n))$
- b. $O(\log(n))$
- c. $O(n)$
- d. $O(n^2)$

Binary Expression Tree

1. Following is a sample implementation of the prefixOrder function, identified the error in the code

```
string ExprTree::prefixOrder(const ExprTree & t){  
    string str = "";  
    str += t->toString() + " ";  
    str += prefix(t->getLeftChild());  
    str += prefix(t->getRightChild());  
    return str;  
}
```

Hashing

1. Given a Hash Function $H(K) = K \% N$, N = size of the array

What would the hashmap look like when inserting 789, 456, 123, 147?

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Given implementing Linear Probing to deal with collision, using the same hash function from the previous question, what would the hashmap look like when inserting 789, 456, 123, 147, 43, 89, 19, 76?

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Given implementing Chaining to deal with collision, using the same hash function from the previous question, what would the hashmap look like when inserting 789, 456, 123, 147, 43, 89, 19, 76?

Diagram illustrating a 10-dimensional vector space. The space is represented by a grid of 10 columns and 5 rows. Each column is labeled with an index from 0 to 9 at the bottom. To the left of each column is a pair of square brackets, indicating that each column represents a vector in the space.