

6장. 데이터 구조 및 알고리즘

Tree

목차

- ▶ Tree as an Abstract Data Type and Structure
- ▶ Terminologies of Tree Structure
- ▶ Characteristics of Tree
- ▶ Binary Search Tree and Implementation
- ▶ Insert and Search Operation of Binary Search Tree
- ▶ Delete Operation and Minimum & Maximum of Binary Search Tree
- ▶ Tree Traversing

1. Tree as an Abstract Data Type and Structure

Tree as an abstract data type

◆ Tree structure

- ◆ Abstract data type
- ◆ Data stored
- ◆ As a tree structure
- ◆ Operations
 - ◆ Ordinary data structure operations just as linked lists
 - ◆ Insert
 - ◆ Delete
 - ◆ Search
- ◆ Special searching approaches for trees and networks
 - ◆ Traverse

① ② ③

①

② ③

④ ⑤ ⑥ ⑦

- Up - side -down 형태
- 컴퓨터에서는 아래의 그림과 같이 표현할 수 있다.
- 위에서부터 뻗어나와 다양한 잎을 가지는 구조
- 데이터는 tree structure 형식으로 저장이 된다.
- Tree structure의 기능은 linked list로 insert, delete, search이다.
- Tree의 특별한 접근이 있는데 모든 데이터를 꺼내는것이 traversing이다.
- Liked list랑은 다르다.

1. Tree as an Abstract Data Type and Structure

Why do we use trees?

- ◇ Because the structure of trees is a good analogy to the various real world structures

- ◇ Corporate structures
- ◇ Group bank accounts
- ◇ Command and control structures

- ◇ Why is the structure one of the most favorite structures?

- ◇ A clear approach of *Divide and Conquer*

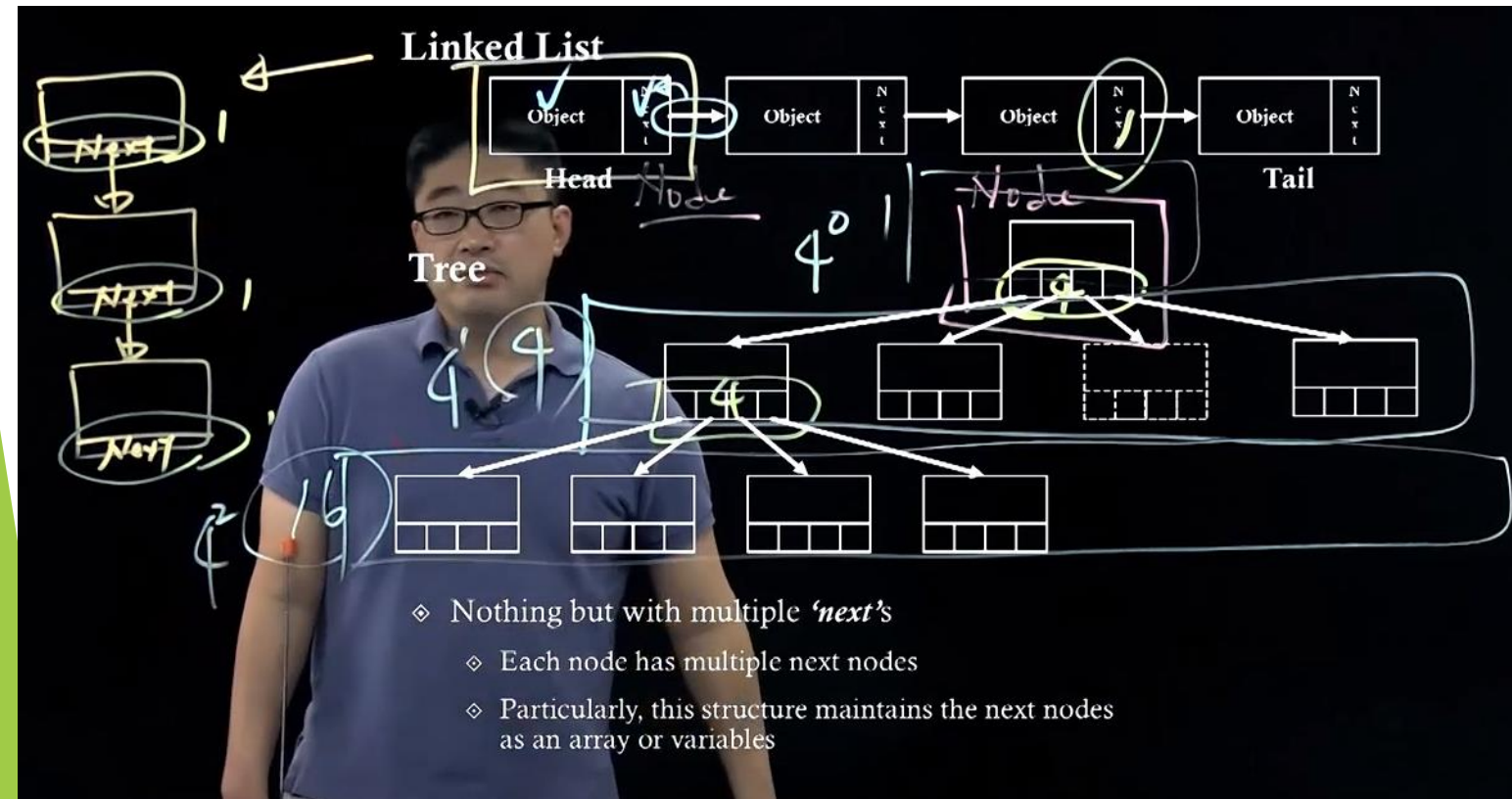


- Tree구조는 현실에 많이 존재
- 어떤 조직의 조직도
- 어떤 회사의 그룹 bank account
- 회사는 개인과는 다르게 체계적인 bank account가 있을 수 있다.
- 조직도나 지휘체계도 있다.
- 이런것들은 divide and conquer를 할 수 있다.
- 저장하는 데이터에 대해서 divide하고 저장한 문제를 풀어나가는 구조
- divide and conquer 적용이 되어있다.
- 또한 recursion 구조를 많이 적용한다.



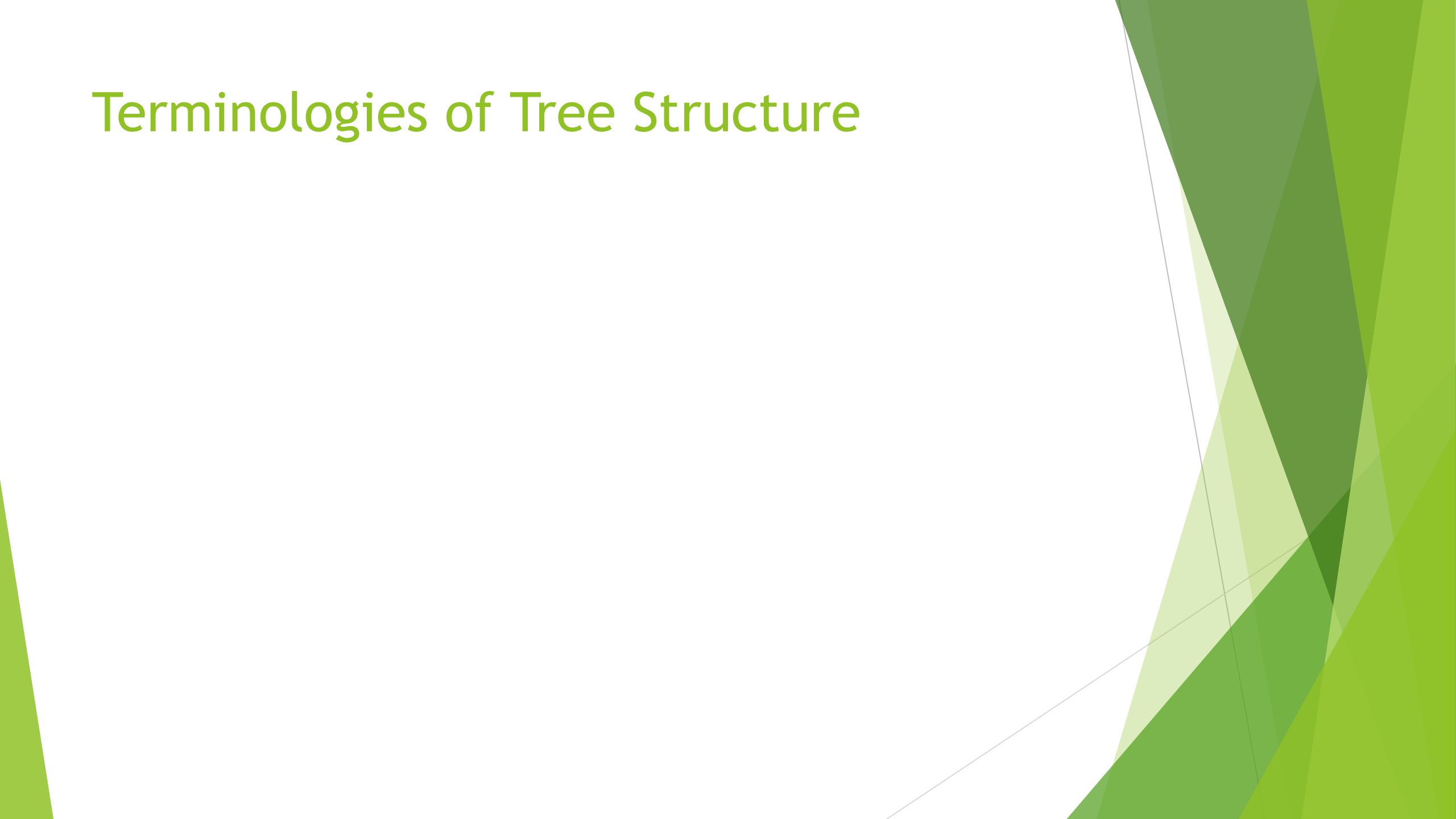
1. Tree as an Abstract Data Type and Structure

Structure of stored data



- Linked list는 Object를 저장하는 레퍼런스 하나
- 다음 node가 무엇인지 레퍼런스 하나로 이루어져 있다.
- tree구조와 유한데 tree 노드는 특징점이 next가 여러 개로 나뉜다.
- Linked list는 next 노드가 하나만 되지만 tree는 여러 개다.
- Tree 구조는 그림에서 점점 커지게 된다.

Terminologies of Tree Structure



Characteristics of Tree



Binary Search Tree and Implementation



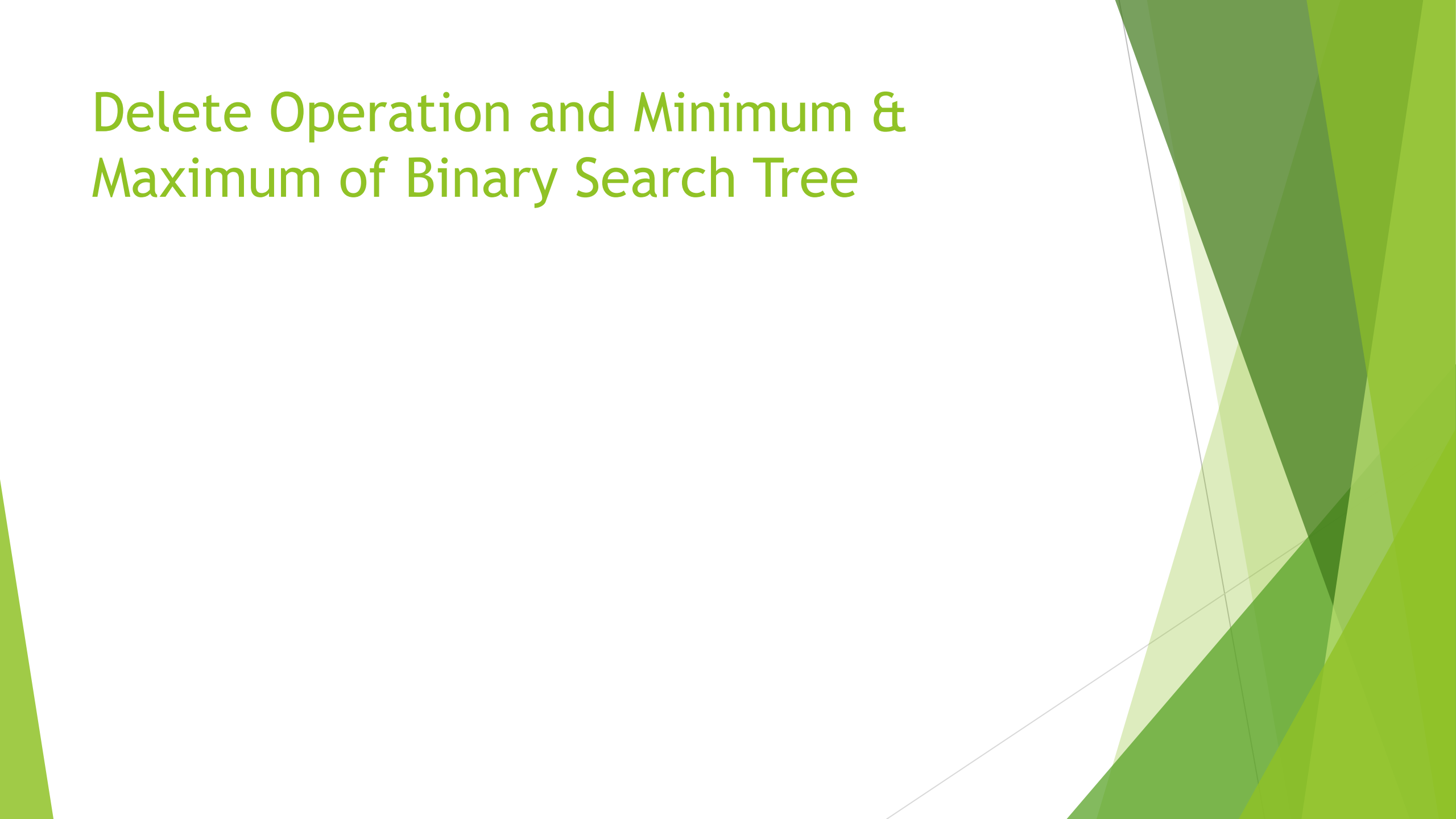
Insert and Search Operation of Binary Search Tree



Insert and Search Operation of Binary Search Tree



Delete Operation and Minimum & Maximum of Binary Search Tree



Tree Traversing

