

# 6 Lecture - CS301

## Important Subjective

1. **What is a stack, and how does it follow the LIFO principle?**

Answer: A stack is an abstract data type that represents a collection of elements with two main operations: push, which adds an element to the top of the stack, and pop, which removes the top element from the stack. It follows the Last-In-First-Out (LIFO) principle, where the last element added is the first to be removed.

2. **How can you implement a stack using an array? What are the advantages and disadvantages of this approach?**

Answer: A stack can be implemented using an array by maintaining a variable to keep track of the topmost element's index. Advantages of this approach include constant time complexity for push and pop operations and efficient use of memory. Disadvantages include fixed size and difficulty in dynamic resizing.

3. **How can you implement a stack using a linked list? What are the advantages and disadvantages of this approach?**

Answer: A stack can be implemented using a linked list by using the head of the list to represent the topmost element. Advantages of this approach include dynamic resizing, efficient use of memory, and easy implementation. Disadvantages include slower access time than an array-based implementation.

4. **What is the purpose of the peek operation in a stack?**

Answer: The peek operation allows you to view the topmost element in the stack without removing it.

5. **What happens when you try to pop an element from an empty stack?**

Answer: When you try to pop an element from an empty stack, an error message is displayed.

6. **What is the time complexity of push and pop operations in a stack implemented using a linked list?**

Answer: The time complexity of push and pop operations in a stack implemented using a linked list is  $O(1)$ .

7. **How can stacks be used to evaluate postfix expressions?**

Answer: Stacks can be used to evaluate postfix expressions by iterating over the expression and performing operations based on the current element. When an operand is encountered, it is pushed onto the stack. When an operator is encountered, the two most recent operands are popped from the stack, and the operation is performed, with the result being pushed back onto the stack.

8. **What is a potential application of stacks in checking for balanced parentheses in an expression?**

Answer: A stack can be used to check for balanced parentheses in an expression by pushing opening parentheses onto the stack and popping them off when a closing parenthesis is encountered. If the stack is empty at the end of the expression, then the parentheses are

balanced.

9. **What is a potential disadvantage of using an array to implement a stack?**

Answer: A potential disadvantage of using an array to implement a stack is that the size is fixed and cannot be dynamically resized.

10. **What is the time complexity of searching for an element in a stack implemented using an array?**

Answer: The time complexity of searching for an element in a stack implemented using an array is  $O(n)$ , as you need to iterate over each element to find the desired one.