

Lecture 4.29

4.6 Overview of doubly linked list, 4.7 Applications of linked list

Agenda

- - Topic introduction
- - Core concepts
- - Code examples
- - Summary

Detailed Concept 1

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 1
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 2

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 2
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 3

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 3
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 4

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 4
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 5

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 5
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 6

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 6
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 7

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 7
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Detailed Concept 8

- Detailed explanation for 4.6 Overview of doubly linked list, 4.7 Applications of linked list part 8
- - Memory management aspects
- - Time complexity considerations

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None
```

Summary

- - Recapped 4.6 Overview of doubly linked list, 4.7 Applications of linked list
- - Discussed node structure and time complexities

Next Lecture Preview

- In the next lecture, we will dive deeper into related operations.