

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 21강 - 양방향 연결 리스트

학습 목표

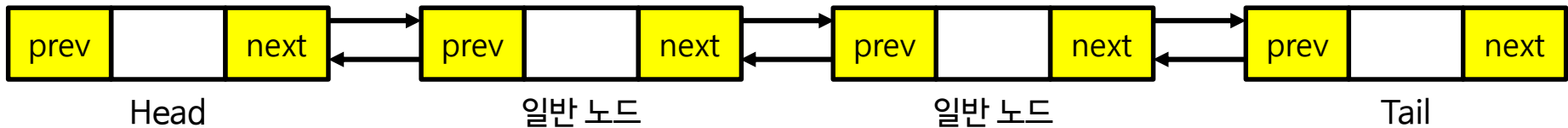
양방향 연결 리스트

- 1) 양방향 연결 리스트의 동작 원리와 구현 방법에 대해서 공부합니다.
- 2) 양방향 연결 리스트를 직접 구현해보며 포인터 구현 능력을 향상시킵니다.

양방향 연결 리스트

양방향 연결 리스트

- 1) 양방향 연결 리스트는 머리(Head)와 꼬리(Tail)를 모두 가진다는 특징이 있습니다.
- 2) 양방향 연결 리스트의 각 노드는 앞 노드와 뒤 노드의 정보를 모두 저장하고 있습니다.
- 3) 우리는 데이터를 '오름차순'으로 저장하는 양방향 연결 리스트를 구현해 볼 것입니다.



양방향 연결 리스트

양방향 연결 리스트 선언하기

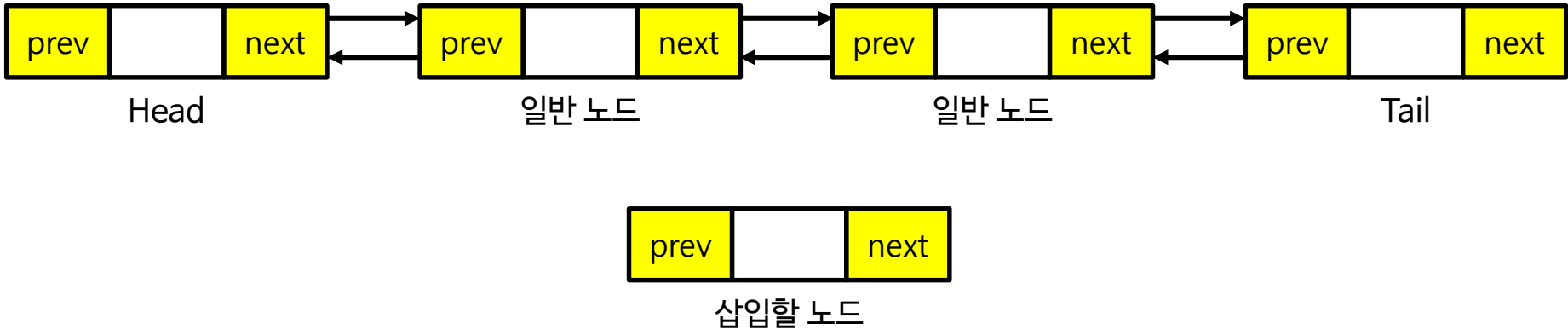
```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int data;
    struct Node *prev;
    struct Node *next;
} Node;

Node *head, *tail;
```

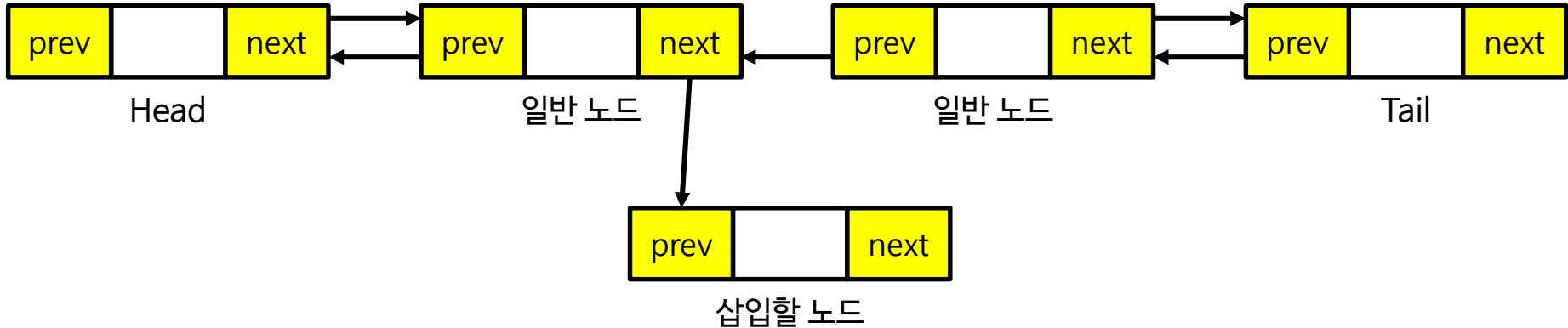
연결 리스트

연결 리스트 삽입 과정 1



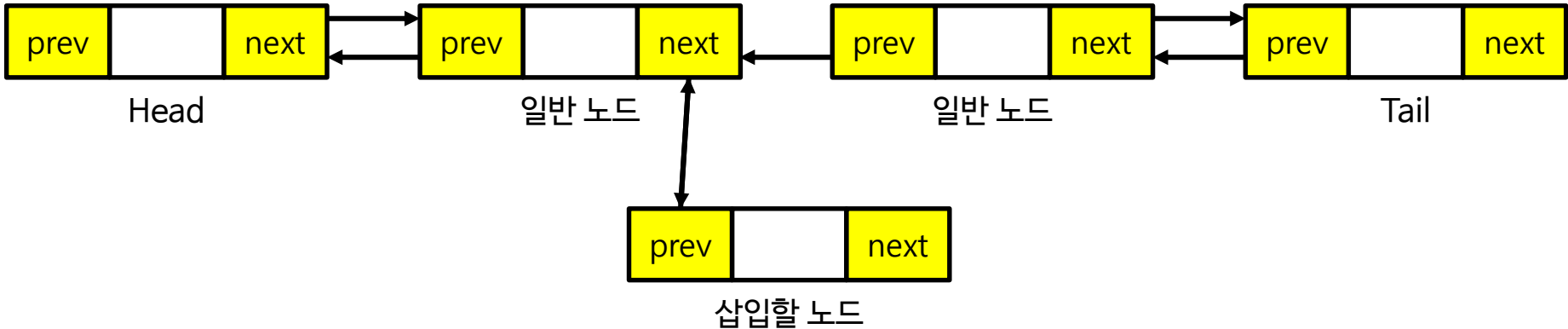
연결 리스트

연결 리스트 삽입 과정 2



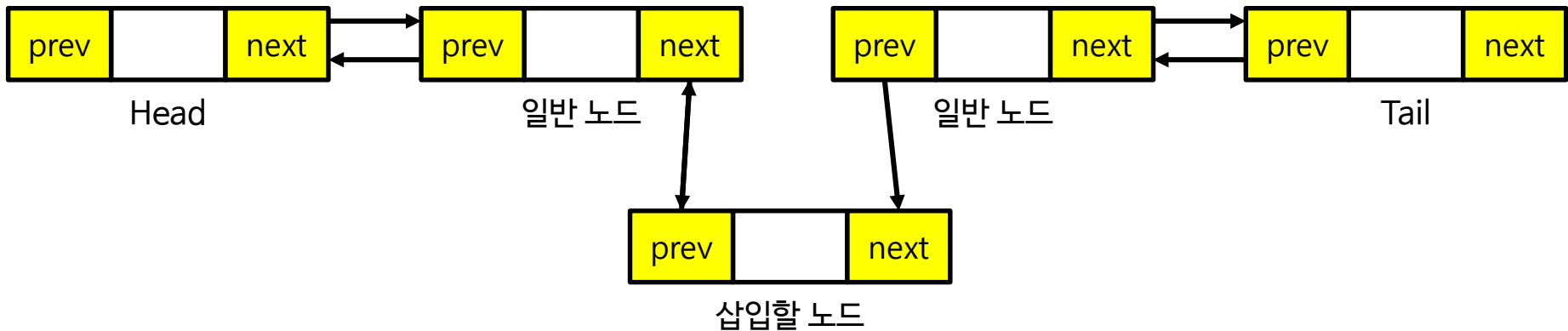
연결 리스트

연결 리스트 삽입 과정 3



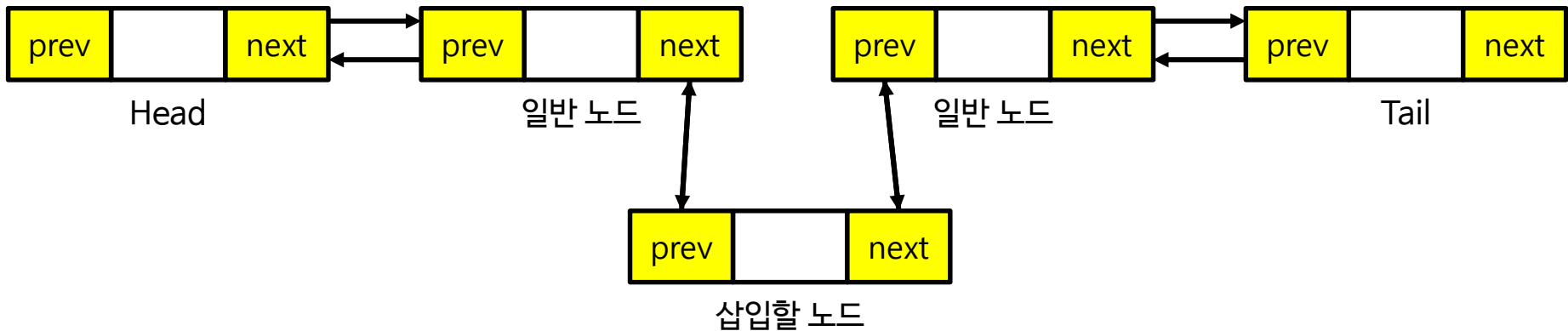
연결 리스트

연결 리스트 삽입 과정 4



연결 리스트

연결 리스트 삽입 과정 5



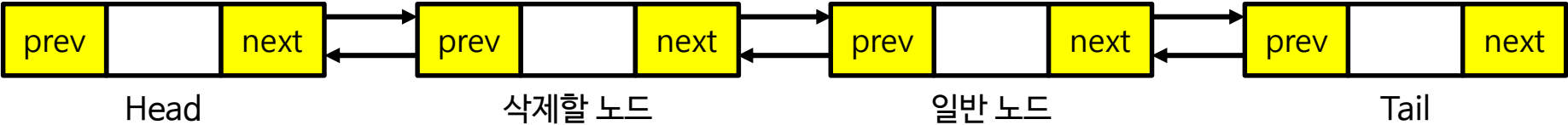
양방향 연결 리스트

양방향 연결 리스트 삽입 함수

```
void insert(int data) {  
    Node* node = (Node*) malloc(sizeof(Node));  
    node->data = data;  
    Node* cur;  
    cur = head->next;  
    while (cur->data < data && cur != tail) {  
        cur = cur->next;  
    }  
    Node* prev = cur->prev;  
    prev->next = node;  
    node->prev = prev;  
    cur->prev = node;  
    node->next = cur;  
}
```

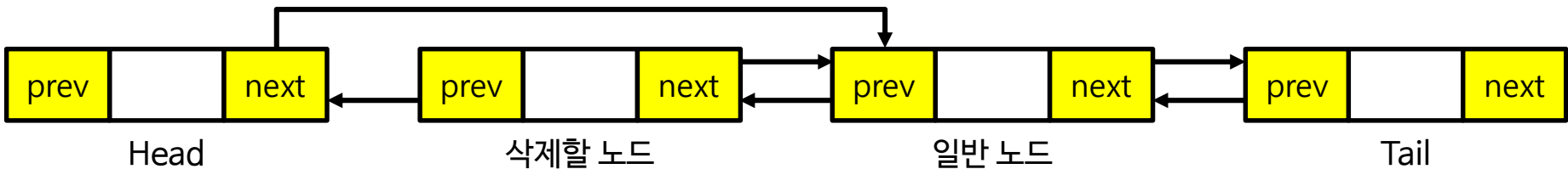
연결 리스트

연결 리스트 삭제 과정 1



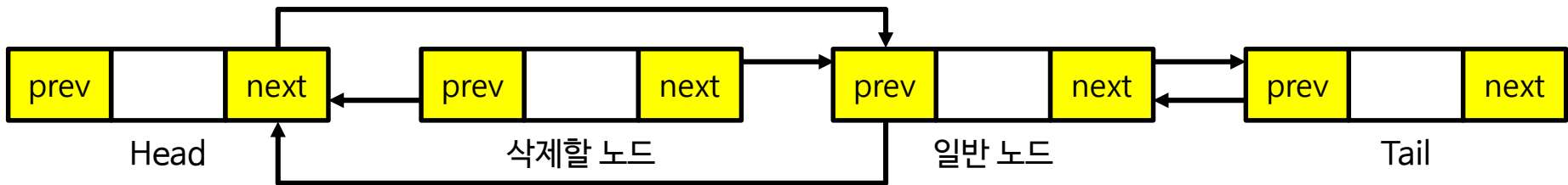
연결 리스트

연결 리스트 삭제 과정 2



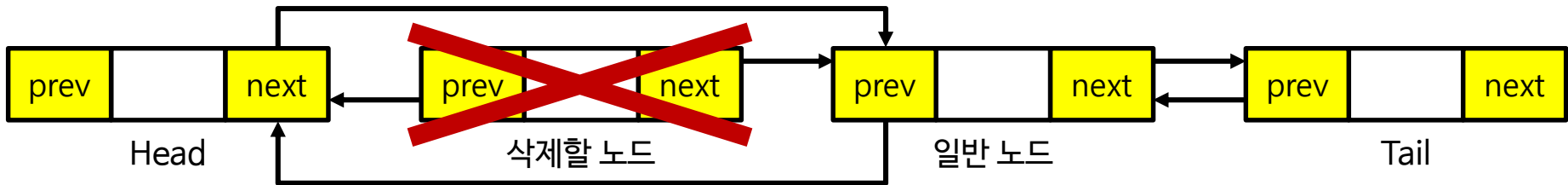
연결 리스트

연결 리스트 삭제 과정 3



연결 리스트

연결 리스트 삭제 과정 4



양방향 연결 리스트

양방향 연결 리스트 삭제 함수

```
void removeFront() {  
    Node* node = head->next;  
    head->next = node->next;  
    Node* next = node->next;  
    next->prev = head;  
    free(node);  
}
```

양방향 연결 리스트

양방향 연결 리스트 전체 출력 함수

```
void show() {  
    Node* cur = head->next;  
    while (cur != tail) {  
        printf("%d ", cur->data);  
        cur = cur->next;  
    }  
}
```


양방향 연결 리스트

완성된 양방향 연결 리스트 사용하기

```
int main(void) {  
    head = (Node*) malloc(sizeof(Node));  
    tail = (Node*) malloc(sizeof(Node));  
    head->next = tail;  
    head->prev = head;  
    tail->next = tail;  
    tail->prev = head;  
    insert(2);  
    insert(1);  
    insert(3);  
    insert(9);  
    insert(7);  
    removeFront();  
    show();  
    system("pause");  
    return 0;  
}
```

양방향 연결 리스트

양방향 연결 리스트 구현에 있어서 주의할 점

- 1) 위 소스코드에 덧붙여서 삽입 및 삭제 기능에서의 예외 사항을 처리할 필요가 있습니다.
- 2) 더 이상 삭제할 원소가 없는데 삭제하는 경우 등을 체크해야 합니다.

배운 내용 정리하기

양방향 연결 리스트

- 1) 양방향 연결 리스트에서는 각 노드가 앞 노드와 뒤 노드의 정보를 저장하고 있습니다.
- 2) 양방향 연결 리스트를 이용하면 리스트의 앞에서부터 혹은 뒤에서부터 모두 접근할 수 있습니다.