6주차 3차시 다항식 연결 자료구조

【학습목표】

1. 연결 자료구조를 이용한 다항식의 덧셈 연산 방법에 대하여 설명할 수 있다.
2. 다항식 연결 자료구조의 삽입 및 덧셈연산을 구분할 수 있다.

학습내용1 : 다항식의 연결 자료구조

1. 다항식의 연결 자료구조 표현

* 단순 연결 리스트를 이용하여 다항식을 표현
- 다항식의 항 : 단순 연결 리스트의 노드

* 노드구조

[그림 5-91] 다항식 노드

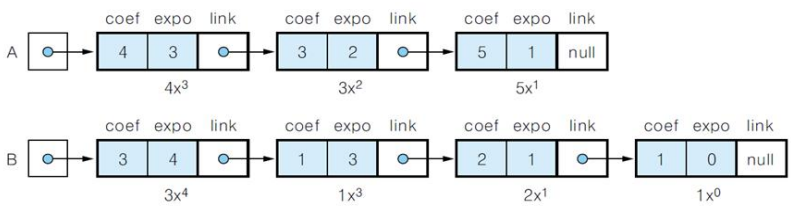
- 각 항에 대해서 계수와 지수를 저장
  - 계수를 저장하는 coef와 지수를 저장하는 expo의 두 개 필드로 구성
  - 링크 필드는 다음 항을 연결하는 포인터로 구성

* 노드에 대한 구조체 정의

```
typedef struct Node{
        float coed;
        int   expo;
        struct Node *link;
}
```

2. 다항식의 단순 연결 리스트 표현

* 다항식 $A(x)=4x^3+3x^2+5x$
* 다항식 $B(x)=3x^4+x^3+2x+1$



[그림 5-92] 다항식 A(x)와 B(x)의 단순 연결 리스트 표현

학습내용2 : 다항식 연결 자료구조의 삽입 연산

1. 다항식 연결 자료구조의 항 삽입 연산

* 다항식에 항을 추가하는 알고리즘

- 다항식 리스트 포인터 PL과 coef 필드값을 저장한 변수 coef, expo 필드값을 저장한 변수 expo 그리고 리스트 PL의 마지막 노드의 위치를 지시하는 포인터 last를 변수로 사용

알고리즘 5-13 다항식에 항을 추가하는 알고리즘

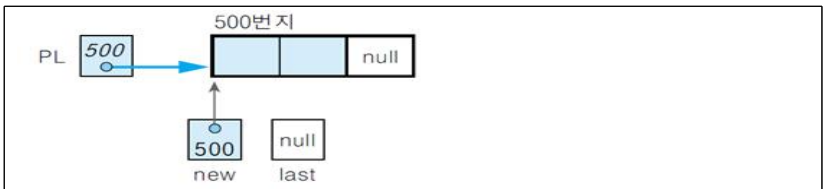
```
appendTerm(PL, coef, expo, last)
    new ← getNode();
    new.expo ← expo;
    new.coef ← coef;
    new.link ← null;
    if (PL = null) then {          // ❶ 공백 리스트인 경우
        PL ← new;                  // ❶-ⓐ
        last ← new;                // ❶-ⓑ
    }
    else {                         // ❷ 공백 리스트가 아닌 경우
        last.link ← new;           // ❷-ⓐ
        last ← new;                // ❷-ⓑ
    }
end appendTerm()
```

❶ [공백 다항식에서의 항 삽입 연산]

[그림 5-93] appendTerm( ) 함수 수행 과정: 알고리즘 설명 ❶

❶-ⓐ PL ← new;

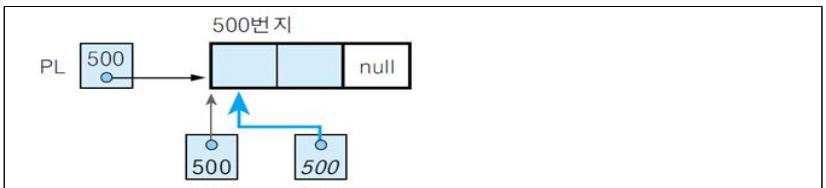
- 포인터 new의 값을 리스트 포인터 PL에 저장하여 노드 new가 리스트 PL의 첫 번째 노드가 되도록 연결



[그림 5-94] appendTerm( ) 함수 수행 과정: 알고리즘 설명 ❶-ⓐ PL ← new;

❶-ⓑ last ← new;

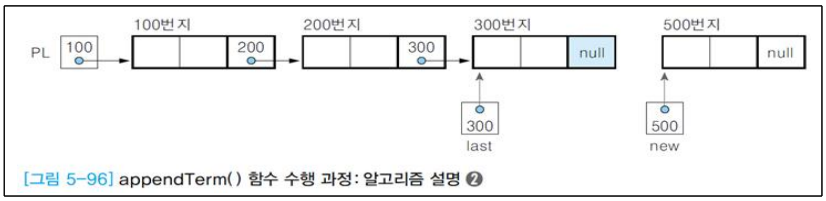
- 포인터 new의 값을 포인터 last에 저장하여 포인터 last가 리스트 PL의 마지막 노드인 노드 new를 가리키도록 지정한다



[그림 5-95] appendTerm( ) 함수 수행 과정: 알고리즘 설명 ❶-ⓑ last ← new;

❷ [다항식에서의 항 삽입 연산]

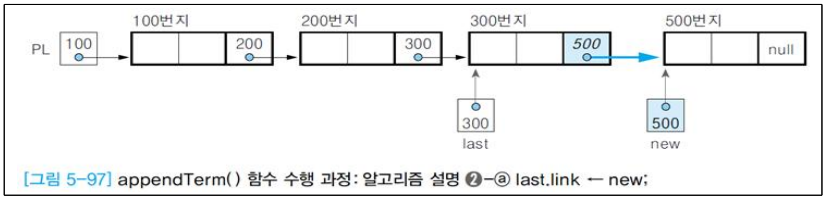
- 새 노드 new를 리스트 PL의 마지막 노드로 삽입한다



[그림 5-96] appendTerm( ) 함수 수행 과정: 알고리즘 설명 ❷

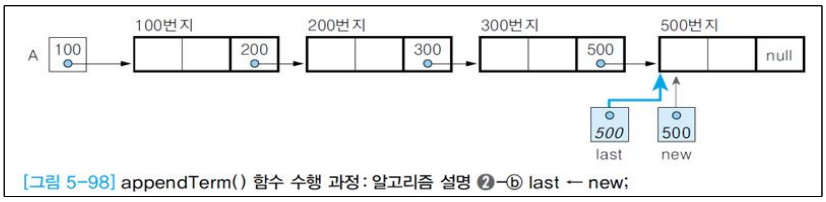
❷-ⓐ last.link ← new;

- 포인터 new의 값을 노드 last의 링크에 저장하여 노드 new를 노드 last의 다음 노드로 연결한다



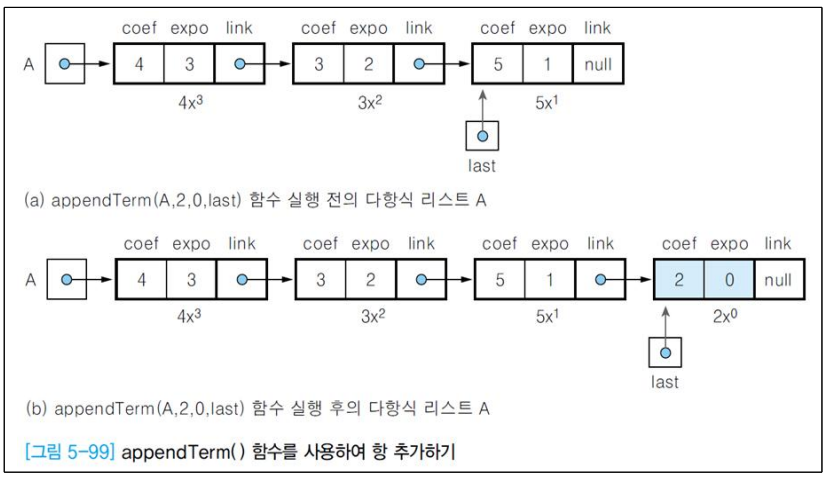
[그림 5-97] appendTerm( ) 함수 수행 과정: 알고리즘 설명 ❷-ⓐ last.link ← new;

❷-ⓑ last ← new;
- 포인터 new의 값을 노드 last에 저장하여 노드 new를 리스트 PL의 마지막 노드로 설정한다



[그림 5-98] appendTerm( ) 함수 수행 과정 : 알고리즘 설명 ❷-ⓑ last ← new;

* 다항식 리스트 A에 appendTerm() 알고리즘을 사용하여 $2x^0$ 추가



[그림 5-99] appendTerm( ) 함수를 사용하여 항 추가하기

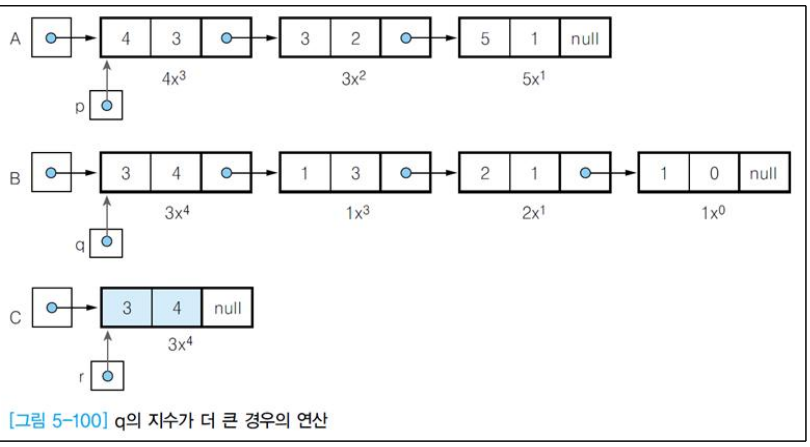
학습내용3 : 다항식 연결 자료구조의 덧셈연산

1. 다항식의 덧셈 연산

* 덧셈 A(x)+B(x)=C(x)를 단순 연결 리스트 자료구조를 사용하여 연산
- 다항식 A(x)와 B(x), C(x)의 항을 지시하기 위해서 세 개의 포인터를 사용
- 포인터 p : 다항식 A(x)에서 비교할 항을 지시
- 포인터 q : 다항식 B(x)에서 비교할 항을 지시
- 포인터 r : 덧셈연산 결과 만들어지는 다항식 C(x)의 항을 지시

2. p.expo < q.expo : 다항식 A(x) 항의 지수가 작은 경우

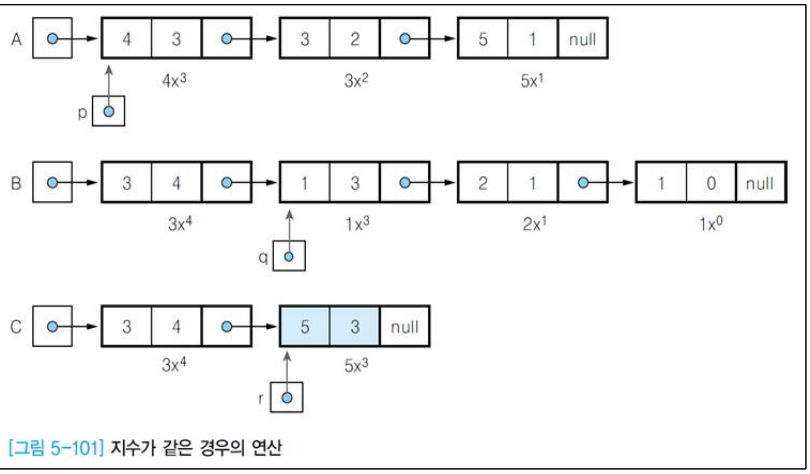
- 두 다항식의 지수가 서로 다른 경우 계수의 덧셈을 할 수 없고 지수가 높은 항부터 나열
- 포인터 q가 가리키는 다항식 B(x)의 항을 C(x)항으로 복사
- q가 가리키는 항에 대한 처리가 끝났으므로 q를 다음 노드로 이동



[그림 5-100] q의 지수가 더 큰 경우의 연산

3. p.expo = q.expo : 두 항의 지수가 같은 경우

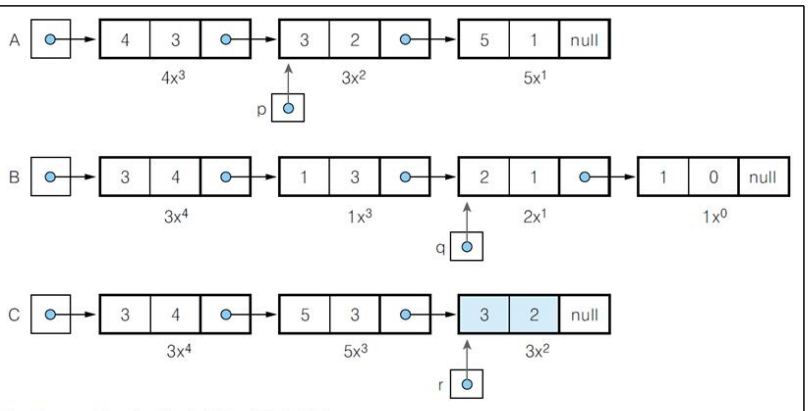
- p.coef 와 q.coef를 더하여 C(x)의 항인 r.coef에 저장하고 지수는 같으므로 p.expo(또는 q.expo)를 r.expo에 저장
- 다음 항을 비교하기 위하여 p, q를 각각 다음 노드로 이동



[그림 5-101] 지수가 같은 경우의 연산

4. p.expo > q.expo : 다항식 A(x) 항의 지수가 큰 경우

- p가 가리키는 다항식 A(x)의 항의 지수가 더 큰 경우에는 포인터 p가 가리키는 항을 C(x)의 항으로 복사
- 포인터 p에 대한 처리가 끝났으므로 p를 다음 노드로 이동



[그림 5-102] p의 지수가 더 큰 경우의 연산

5. 다항식의 덧셈 알고리즘

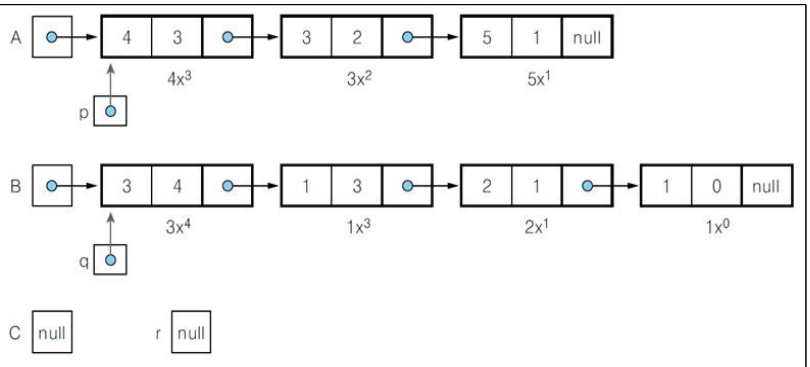
알고리즘 5-14 단순 연결 리스트를 이용한 다항식의 덧셈 알고리즘

```
addPoly(A, B)
    // 단순 연결 리스트로 표현된 다항식 A와 B를 더하여
    // 새로운 다항식 C를 반환
    p ← A;
    q ← B;
    C ← null;      // 결과 다항식
    r ← null;      // 결과 다항식의 마지막 노드를 지시
    while (p ≠ null and q ≠ null) do {      // p, q는 순회 포인터
        case {
            p.expo = q.expo :
                        sum ← p.coef + q.coef
                        if (sum ≠ 0) then appendTerm(C, sum, p.expo, r);
                        p ← p.link;
                        q ← q.link;
            p.expo < q.expo :
                        appendTerm(C, q.coef, q.expo, r);
                        q ← q.link;
```

```
            else :      // p.expo > q.expo인 경우
                        appendTerm(C, p.coef, p.expo, r);
                        p ← p.link;
            } // end case
        } // end while

        while (p ≠ null) do {      // A의 나머지 항들을 복사
                appendTerm(C, p.coef, p.expo, r);
                p ← p.link;
        }
        while (q ≠ null) do {      // B의 나머지 항들을 복사
                appendTerm(C, q.coef, q.expo, r);
                q ← q.link;
        }
        return C;
    end addPoly()
```

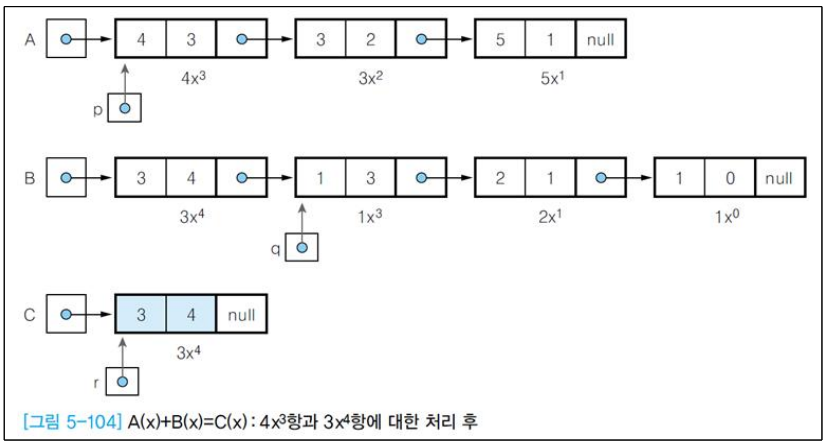
* 다항식 덧셈 예제
- $A(x)=4x^3+3x^2+5x$
- $B(x)=3x^4+x^3+2x+1$



[그림 5-103] A(x)+B(x)=C(x) : 초기 상태

① $4x^3$항과 $3x^4$항에 대한 처리

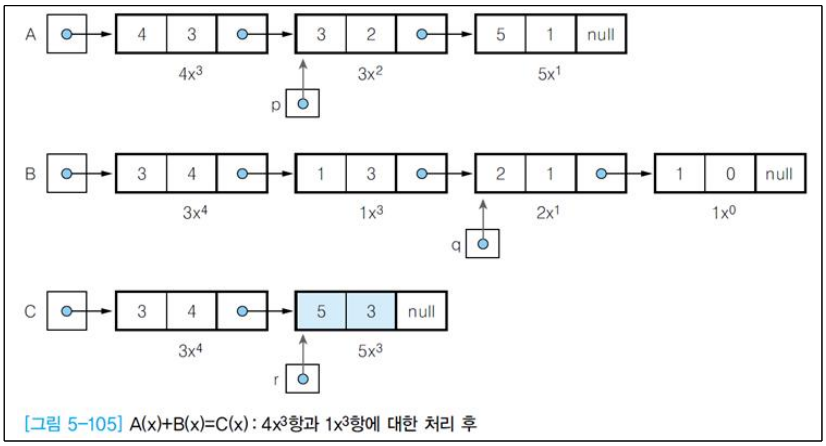
- p.expo < q.expo 이므로 지수가 큰 q를 리스트 PL에 복사
- 포인터 q는 다음 노드로 이동



[그림 5-104] A(x)+B(x)=C(x) : $4x^3$항과 $3x^4$항에 대한 처리 후

② $4x^3$항과 $1x^3$ 항에 대한 처리

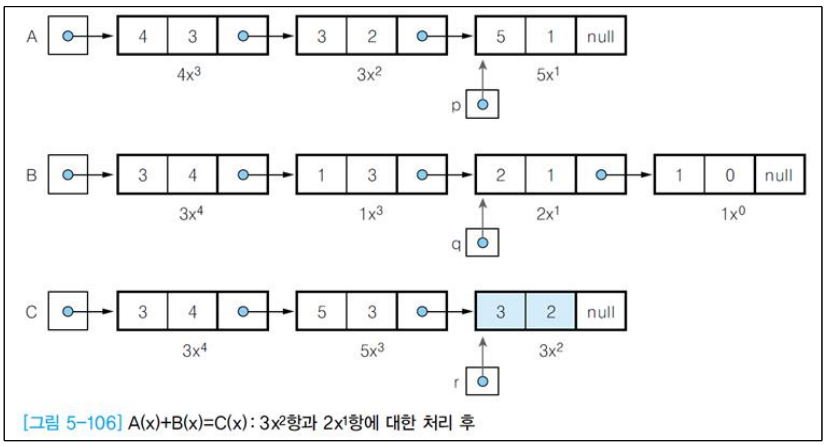
- p.expo = q.expo 이므로 두 노드의 coef 펄드의 값을 더하고 expo 필드의 값을 복사한 노드($5x^3$항)를 리스트 C에 추가
- 포인터 p, q는 다음 노드로 이동



[그림 5-105] A(x)+B(x)=C(x) : $4x^3$항과 $1x^3$항에 대한 처리 후

③ $3x^2$항과 $2x^1$ 항에 대한 처리

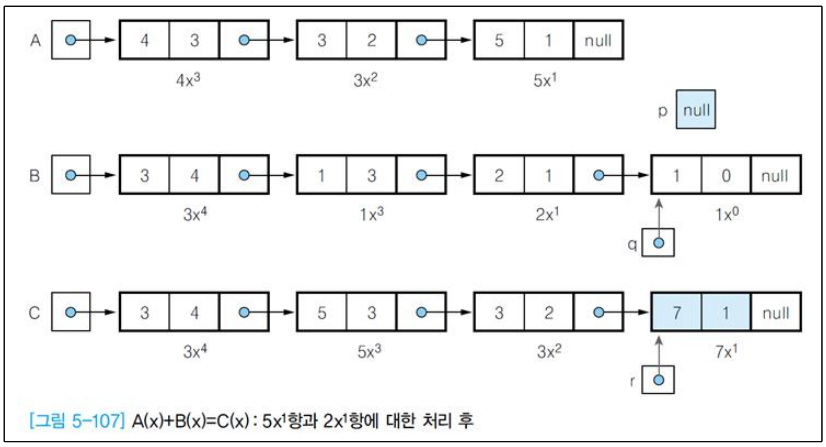
- p.expo > q.expo 이므로 지수가 큰 p 노드를 리스트 C에 복사
- 포인터 p는 다음 노드로 이동



[그림 5-106] A(x)+B(x)=C(x) : $3x^2$항과 $2x^1$항에 대한 처리 후

④ $5x^1$항과 $2x^1$ 항에 대한 처리

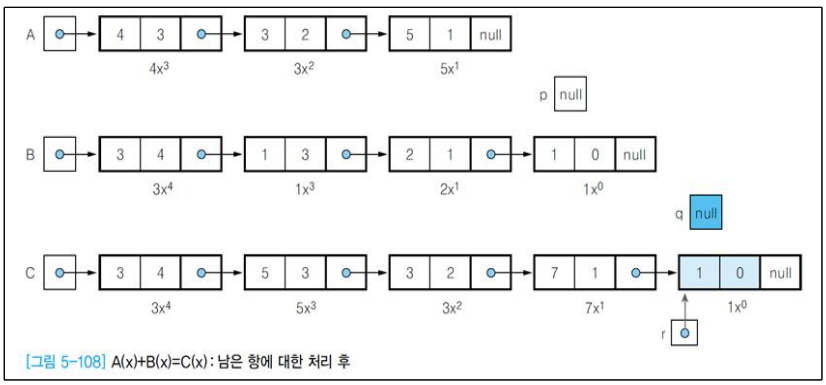
- p.expo = q.expo이므로 두 노드의 coef 필드의 값을 더하고 expo 필드의 값을 복사한 노드($7x^1$항)를 리스트 C에 추가
- 포인터 p, q는 다음 노드로 이동



[그림 5-107] A(x)+B(x)=C(x) : $5x^1$항과 $2x^1$항에 대한 처리 후

⑤ B(x)의 남은 항에 대한 처리

- 포인터 p가 null이므로 다항식 A(x)의 항에 대한 처리 끝
- 처리할 항이 남아있는 다항식 B(x)의 포인터 q는 null이 될 때까지 모든 노드를 복사하여 리스트 C에 추가



[그림 5-108] A(x)+B(x)=C(x) : 남은 항에 대한 처리 후

【학습정리】

1. 단순 연결 리스트를 이용하여 다항식을 표현할 수 있다.

2. 다항식에 있는 하나의 항은 하나의 노드로 표현할 수 있다.

3. 노드는 각 항에 대해서 계수와 지수를 저장하기 위해 데이터 필드 두 개와 다음 항에 대한 노드를 연결하는 링크 필드로 구성한다.