

# 자료구조 과제 #4

214823 박종현

## 조건

- ✓ LinkedList를 활용하여 SparsePoly 클래스 구현
- ✓ 과제 #3 Polynomial ADT 연산 중 add, equal, degree, coef 구현
- ✓ read, display 구현
- ✓ add 연산:  $p1 + p2$  가능하게 구현
- ✓ equal 연산:  $p1 == p2$  가능하게 구현
- ✓ 교재의 예제 입력과 달리 임의의 차수 순으로 입력 가능해야함
- ✓ 다항식의 노드는 차수가 감소하는 순서를 유지해야 함

## add 연산의 시간복잡도 big-O

- ✓ 다항식의 차수:  $n$ , 계수가 0이 아닌 항의 개수:  $m$
  - ✓ 다항식의 차수  $n$ 에 대해,  $n$ 회 반복 수행함.
- $\therefore O(n)$

/out/out.png 예제 입력 결과

```
shapelayer@ShapeLayer-MacBook:~/Documents/GitHub/ShapeLayer/compute...
28% 8.7 GB 75%
python app.py
계수 차수 입력 (종료 : -1): 3 12
계수 차수 입력 (종료 : -1): 2 8
계수 차수 입력 (종료 : -1): 1 0
계수 차수 입력 (종료 : -1): -1 -1
    입력 다항식 : 3 x^12 + 2 x^8 + 1 x^0
계수 차수 입력 (종료 : -1): 8 12
계수 차수 입력 (종료 : -1): -3 10
계수 차수 입력 (종료 : -1): 10 6
계수 차수 입력 (종료 : -1): -1 -1
    입력 다항식 : 8 x^12 + -3 x^10 + 10 x^6
A = 3 x^12 + 2 x^8 + 1 x^0
B = 8 x^12 + -3 x^10 + 10 x^6
A + B = 11 x^12 + -3 x^10 + 2 x^8 + 10 x^6 + 1 x^0
```

/src/app.py 프로그램 소스

```
1 from typing import TypeVar, Generic
2
3 T = TypeVar("T")
```

Python

```

4
5 class Node(Generic[T]):
6     def __init__(self, data: T, link: 'Node'):
7         self.data: T = data
8         self.link = link
9
10 class LinkedList:
11     def __init__(self):
12         self.head = None
13
14     def is_empty(self):
15         return self.head == None
16
17     def is_full(self):
18         return False
19
20     def get_node(self, pos: int) -> Node:
21         if pos < 0:
22             return None
23         node = self.head
24         while pos > 0 and node != None:
25             node = node.link
26             pos -= 1
27         return node
28
29     def get_entry(self, pos: int) -> T:
30         node = self.get_node(pos)
31         if node == None:
32             return None
33         else:
34             return node.data
35
36     def __getitem__(self, key: int) -> T:
37         return self.get_entry(key)
38
39     def insert(self, pos: int, elem: T):
40         prev = self.get_node(pos - 1)
41         if prev == None:
42             self.head = Node(elem, self.head)
43         else:
44             node = Node(elem, prev.link)
45             prev.link = node
46
47     def delete(self, pos: int):
48         prev = self.get_node(pos - 1)
49         if prev == None:
50             if self.head is not None:
51                 self.head = self.head.link
52             elif prev.link != None:
53                 prev.link = prev.link.link
54
55 class Term:
56     def __init__(self, expo, coef):
57         ...
58         :param expo: 항의 지수

```

```

59         :param coef: 항의 계수
60         ...
61         self.expo = expo
62         self.coef = coef
63
64     class SparsePoly(LinkedList):
65         def __str__(self):
66             coef_buf = self.coef()
67             str_buf = []
68             degs = [*coef_buf.keys()]
69             while degs:
70                 deg = degs.pop()
71                 str_buf.append(f'{coef_buf[deg]} x^{deg}')
72             return ' +\t'.join(str_buf)
73
74         def read(self):
75             ...
76             사용자 입력 함수
77             ...
78             while True:
79                 gets = [*map(int, input('계수 차수 입력(종료: -1): ').split())]
80                 if gets[0] == -1 and gets[1] == -1:
81                     break
82                 if (len(gets) != 2):
83                     # fallback
84                     continue
85                 c, d = gets
86                 if self.get_node(d) == None:
87                     self._fill_until(d)
88                 self.get_node(d).data = c
89
90             self.display()
91
92         def _fill_until(self, pos: int):
93             node = self.head # pos 0
94             if node == None:
95                 self.head = Node(0, None)
96                 node = self.head
97             for i in range(pos):
98                 if node.link == None:
99                     node.link = Node(0, None)
100                 node = node.link
101
102         def display(self):
103             ...
104             화면 출력 함수
105             ...
106             print(f'\t입력 다항식: {self.__str__()}')
107
108         def add(self, polyB: 'SparsePoly') -> 'SparsePoly':
109             ...
110             다항식 덧셈 함수
111             ...
112             _new = SparsePoly()
113             _new.head = Node(0, None)

```

```

114     entry_self, entry_other, entry_new = self.head, polyB.head, _new.head
115     while True:
116         if entry_self:
117             entry_new.data += entry_self.data
118             entry_self = entry_self.link
119         if entry_other:
120             entry_new.data += entry_other.data
121             entry_other = entry_other.link
122         if entry_self != None or entry_other != None:
123             entry_new.link = Node(0, None)
124             entry_new = entry_new.link
125         else:
126             break
127     return _new
128
129     def __add__(self, other: 'SparsePoly') -> 'SparsePoly':
130         return self.add(other)
131
132     def equal(self, other: 'SparsePoly') -> bool:
133         '''
134         다항식 항등 평가
135         '''
136         entry_self, entry_other = self.head, other.head
137         while True:
138             if entry_self == None and entry_other == None:
139                 return True
140             if entry_self == None or entry_other == None:
141                 return False
142             if entry_self.data != entry_other.data:
143                 return False
144             entry_self, entry_other = entry_self.link, entry_other.link
145
146     def __eq__(self, other: 'SparsePoly') -> 'SparsePoly':
147         return self.equal(other)
148
149     def degree(self):
150         '''
151         다항식 차수 반환
152         '''
153         i = 0
154         entry = self.head
155         while True:
156             if entry.link == None:
157                 return i
158             i += 1
159             entry = entry.link
160
161     def coef(self) -> dict[int, int]:
162         '''
163         다항식 계수 반환
164         '''
165         coef_buf = {}
166         node = self.head
167         degree = 0
168         while node != None:

```

```

169         if node.data != 0:
170             coef_buf[degree] = node.data
171             node = node.link
172             degree += 1
173         return coef_buf
174
175 if __name__ == '__main__':
176     a, b = SparsePoly(), SparsePoly()
177     a.read()
178     b.read()
179     print(f'\tA = {a}')
180     print(f'\tB = {b}')
181     print(f'\tA + B = {a + b}')
182

```

## /1.in 예제 입력

```

1 3 12
2 2 8
3 1 0
4 -1 -1
5 8 12
6 -3 10
7 10 6
8 -1 -1

```

## /1.out 예제 출력

```

1 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1):   입력 다항식: 3 x^12
+ 2 x^8 + 1 x^0
2 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1): 계수 차수 입력(종료: -1):   입력 다항식: 8 x^12
+ -3 x^10 + 10 x^6
3 A = 3 x^12 + 2 x^8 + 1 x^0
4 B = 8 x^12 + -3 x^10 + 10 x^6
5 A + B = 11 x^12 + -3 x^10 + 2 x^8 + 10 x^6 + 1 x^0

```