

자료구조 과제 3

20211533 남정연

1. 실행 결과

```
Microsoft Visual Studio 디버그 콘솔
+ 32x^33 + 238x^32 + 166x^30 + 155x^29 + 58x^28 + 220x^27 + 49x^26 + 80x^25 + 2x^24 + 83x^23 + 176x^21 + 186x^20 + 54x^19 + 26x^18 + -22x^17 + 34x^16 + 158x^15 + 161x^14 + 144x^13 + 180x^12 + 58x^10 + 90x^9 + 180x^7
C:\Users\Wtonyn\Github\cpp_test\64\Debug\cpp_test.exe(프로세스 14716개)이
(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

2. 코드 설명

```
/**
 * polynomial A(n)의 모든 항에 대하여
 * polynomial B(m)의 모든 항을 곱하고
 * polynomial C에 이를 저장하는 함수
 *
 * 최악의 경우(곱한 모든 항들의 지수가 다름) 1, 2, ..., n*m 번 순차 탐색
 * 따라서 시간 복잡도는  $O(1 + 2 + \dots + nm)$ 임
 *
 *  $O(nm*(nm+1)/2)$ 
 *  $= O(nm^2)$ 
 *  $= O((nm)^2)$ 
 *
 * 최종 시간 복잡도(Worst Case)
 *  $O((nm)^2)$ 
 */
void pmult(poly_pointer A, poly_pointer B, poly_pointer& C) {
    float a;
    int b;
    poly_pointer curA = A;
    poly_pointer curB = NULL;
    poly_pointer curC = NULL;

    // n 번 반복
    while(curA) {
        curB = B;
        // m 번 반복
        while(curB) {
            a = curA->coef * curB->coef;
            b = curA->expon + curB->expon;
            curC = C;
            int flag = 0;
```

```

// 순차 탐색: 1, 2, ..., n*m
while(curC) {
    if(curC->expon == b) {
        curC->coef += a;
        flag = 1;
        break;

        if(curC->link) {
            curC = curC->link;
        } else {
            break;
        }
    }

    // 일치하는 지수가 없으면 맨 뒤에 추가
    if(!flag) {
        poly_pointer node = new poly_node;
        node->coef = a;
        node->expon = b;
        node->link = NULL;
        if(curC) {
            curC->link = node;
        } else {
            C = node;
        }
    }
    curB = curB->link;
}
curA = curA->link;
}
}

// polynomial 을 지수 내림차순으로 정렬하는 함수 (bubble sort)
// O(n^2)
void sortPoly(poly_pointer poly) {
    // bubble sort 구현
    // 구체적인 코드 생략
}

```

가장 time cost가 높은 두 함수 pmult와 sortPoly(bubble sort)의 시간 복잡도가 $O(N^2)$ 이다(N 은 최종적으로 만들어진 다항식의 항의 수). 따라서 A.txt로 입력 받은 다항식의 항의 개수를 m , B.txt로 입력 받은 다항식의 항의 개수를 n 으로 한다면 전체 코드의 시간 복잡도는 $O((nm)^2)$ 이다.