

자료구조 과제 2

20211533 남정연

1. 실행 결과

```
Microsoft Visual Studio 디버그 콘솔
14x^75 + 16x^73 + 20x^70 + 4x^68 + 21x^65 + 12x^64 + 24x^63 + 26x^60 +
2x^59 + 6x^58 + 14x^56 + 16x^55 + 18x^54 + 4x^53 + 12x^50 + 3x^49 + 2
1x^46 + 73x^45 + 62x^43 + 153x^40 + 63x^39 + 78x^38 + 72x^37 + 94x^35
+ 202x^34 + 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\wtynyn\source\repos\CSE3080\Debug\CSE3080.exe(프로세스 17832개)
이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

2. 코드 설명

```
/**
 * polynomial A의 모든 항에 대하여
 * polynomial B의 모든 항을 곱하고
 * polynomial C에 이를 저장하는 함수
 *
 * 최악의 경우(곱한 모든 항들의 지수가 다름) lenC는 1, 2, ..., lenA*lenB 까지
증가
 * 따라서 시간 복잡도는 O(1 + 2 + ... + lenC)임
 *
 * O(lenC*(lenC+1)/2)
 * = O(lenC^2)
 * = O((lenA*lenB)^2)
 *
 * 최종 시간 복잡도(Worst Case)
 * O((lenA*lenB)^2)
 */
void pmult(polynomial* A, polynomial* B, polynomial* C, int lenA, int lenB,
int* lenC) {
    float a;
    int i, j, k, b;
    // lenA 번 반복
    for (i = 0; i < lenA; i++) {
        if (A[i].coef == 0) continue;
        // lenB 번 반복
        for (j = 0; j < lenB; j++) {
            if (B[j].coef == 0) continue;
            a = A[i].coef * B[j].coef;
```

```

        b = A[i].expon + B[j].expon;
        // 계수가 0 이면 건너뛰기
        if(a==0) continue;
        // 순차탐색
        // O(lenC)
        for (k = 0; k < *lenC; k++) {
            if (C[k].expon == b) {
                C[k].coef += a;
                break;
            }
        }
        // 일치하는 지수가 없으면 맨 뒤에 추가
        if(k == *lenC) {
            C[k].expon = b;
            C[k].coef = a;
            (*lenC)++;
        }
    }
}

// polynomial 을 지수 내림차순으로 정렬하는 함수 (bubble sort)
// O(len^2)
void sortPoly(polynomial* poly, int len) {
    int i, j, t, tt;
    // bubble sort 구현
    for(i=0;i<len;i++) {
        for(j=0;j<len;j++) {
            if(poly[i].expon > poly[j].expon) {
                t = poly[i].expon;
                tt = poly[i].coef;
                poly[i].expon = poly[j].expon;
                poly[i].coef = poly[j].coef;
                poly[j].expon = t;
                poly[j].coef = tt;
            }
        }
    }
}

```

가장 time cost가 높은 두 함수 pmult와 sortPoly의 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.