Data Structure HW2

41147009S 陳炫佑

Q1

(a)

100
for each arr[i] = v meaning vx^i

(b)

3
for each arr[i] = v meaning vx^i

(c)

4
arr[i] have two element p, v
for each arr[i].p, arr[i].v meaning $arr[i].v * x^a rr[i].p$

Q2

(1)

B^t	**r**	**c**	**v**
$B^t[0]$	5	9	8
$B^t[1]$	0	4	2
$B^t[2]$	0	8	2
$B^t[3]$	1	0	1
$B^t[4]$	1	7	1
$B^t[5]$	2	8	1
$B^t[6]$	3	0	1
$B^t[7]$	3	4	-1
$B^t[8]$	3	7	1

(2)

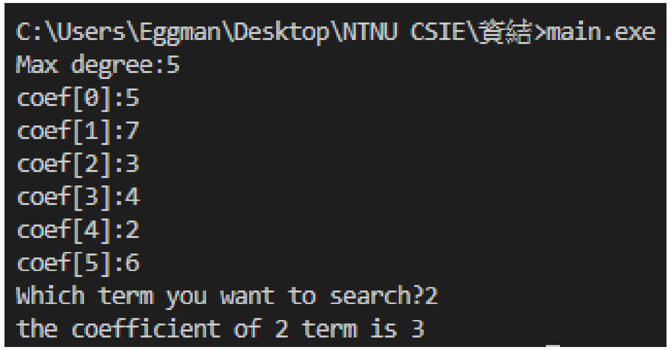
AB	r	c	v
AB[0]	7	5	12
AB[1]	0	1	2
AB[2]	0	3	2
AB[3]	3	0	12
AB[4]	3	2	2
AB[5]	3	3	-4
AB[6]	4	0	2
AB[7]	4	3	-1
AB[8]	5	0	6
AB[9]	5	2	-1
AB[10]	5	3	-4
AB[11]	6	1	-1
AB[12]	6	3	-1

Q3

(1)

```cpp
#include <bits/stdc++.h>
#define MAX_degree 101
using namespace std;
struct polynomial {
    int degree;
    int coef[MAX_degree] = {0};
};
int coeff1(polynomial a, int p) { return a.coef[p]; }
int main() {
    polynomial poly;
    printf("Max degree:");
    scanf("%d", &poly.degree);
    for (int i = 0; i <= poly.degree; ++i) {
        printf("coef[%d]:", i);
        scanf("%d", &poly.coef[i]);
    }
    int p;
    printf("Which term you want to search?");
    scanf("%d", &p);
    printf("the coefficient of %d term is %d", p, coeff1(poly, p));
}
```

result



```
C:\Users\Eggman\Desktop\NTNU CSIE\資結>main.exe
Max degree:5
coef[0]:5
coef[1]:7
coef[2]:3
coef[3]:4
coef[4]:2
coef[5]:6
Which term you want to search?2
the coefficient of 2 term is 3
```

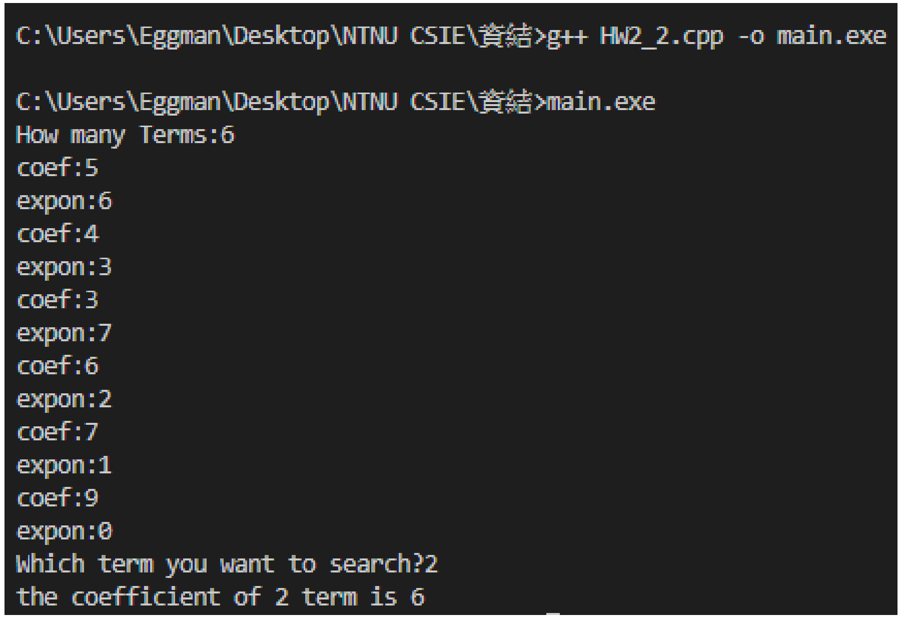
READ ME

```
1.input the polyomial's max degree
2.input every coefficient of the term
3.input the term you want to search
then it will putput the coefficient of the term you search
```

(2)

```cpp
#include <bits/stdc++.h>
#define MAX_TERMS 100
using namespace std;
struct polynomial {
    int coef;
    int expon;
};
int avail = 0;
int coeff2(polynomial *a, int p) {
    for (int i = 0; i < avail; ++i)
        if ((a + i)->expon == p) return (a + i)->coef;
    return 0;
}
int main() {
    polynomial terms[MAX_TERMS];
    printf("How many Terms:");
    scanf("%d", &avail);
    for (int i = 0; i < avail; ++i) {
        printf("coef:", i);
        scanf("%d", &terms[i].coef);
        printf("expon:", i);
        scanf("%d", &terms[i].expon);
    }
    int p;
    printf("Which term you want to search?");
    scanf("%d", &p);
    printf("the coefficient of %d term is %d", p, coeff2(terms, p));
}
```

result



```
C:\Users\Eggman\Desktop\NTNU CSIE\資結>g++ HW2_2.cpp -o main.exe

C:\Users\Eggman\Desktop\NTNU CSIE\資結>main.exe
How many Terms:6
coef:5
expon:6
coef:4
expon:3
coef:3
expon:7
coef:6
expon:2
coef:7
expon:1
coef:9
expon:0
Which term you want to search?2
the coefficient of 2 term is 6
```

READ ME

```
1.input how many terms of this polyomial
2.input every term's coefficient and exponent
3.input the term you want to search
then it will putput the coefficient of the term you search
```

(3)

the time complexity of (1) is O(1)

`because the method just return the value of this term.`

the time complexity of (2) is O(n)

because the method need to find where the term are in the array

(4)

the time complexity of representation I to implement polynomial addition is O(m)

because the addition just need to add every term of polynomial2 to polynomial1

the time complexity of representation II to implement polynomial addition is O(mn)

because the addition need to search every locate of polynomial2's term on polynomial1, and add it