9/1/22

* Section 10 - Sparse Matrix and Polynomial Representation

159. Sparse Matrix Representation

	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	3	0
2	0	0	8	0	0	10	0	0	0
3	0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	2	0	0	0	0	0	0
7	0	0	0	6	0	0	0	0	0
8	0	9	0	0	5	0	0	0	0

8 x 9

72 x 2 = 144 bytes

↑ int

1. Coordinate List / 3-Column Representation
2. Compressed Sparse Row

* 3-Column Representation

row	column	element
8	9	8
1	8	3
2	3	8
2	6	10
4	1	4
6	3	2
7	4	6
8	2	9
8	5	5

* Compressed Sparsed rows

A [3, 8, 10, 4, 2, 6, 9, 5] // Non-zero elements

IA [0, 1, 3, 3, 4, 4, 5, 6, 8] // row elements count in each row + previous count
(0 1 2 3 4 5 6 7 8)

JA [8, 3, 6, 1, 3, 4, 2, 5] // column element corresponding to row element

8+9+8 = 25 × 2 = 50 bytes
↑ int

160. Addition of Sparse Matrices
Size must be same of Matrices

	1	2	3	4	5	6
1	0	0	0	6	0	0
2	0	7	0	0	0	0
3	0	2	0	5	0	0
4	0	0	0	0	0	0
5	4	0	0	0	0	0

A=

	0	1	2	3	4	5	
A	5	1	2	3	3	5	row
	6	4	2	2	4	1	col
	5	6	7	2	5	4	el.

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	3	0	0	5	0
3	0	0	2	0	6	7
4	0	0	0	9	0	0
5	8	0	0	0	0	0

B=

	0	1	2	3	4	5	6
B	5	2	2	3	3	4	5
	6	2	5	3	6	4	1
	6	3	5	2	7	9	8

$A+B=C=$

	1	2	3	4	5	6
1	0	0	0	6	0	0
2	0	10	0	0	5	0
3	0	2	2	5	0	7
4	0	0	0	9	0	0
5	12	0	0	0	0	0

C

	0	1	2	3	4	5	6	7	8	9
	5	1	2	2	3	3	3	3	4	5
	6	4	2	5	2	3	4	6	4	1
	9	6	10	5	2	2	5	7	9	12

161. Array Representation of Sparse Matrix

$$A = \begin{bmatrix} 0 & 0 & 7 & 0 & 0 \\ 2 & 0 & 0 & 5 & 0 \\ 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

4 × 5

A	0	1	2	3	4	5
i	4	1	2	2	3	4
j	5	3	1	4	1	5
x	5	7	2	5	9	4

S

m	4
n	5
num	5
e	

	0	1	2	3	4	5
i						
j						
x						

```
struct Element {
    int i;
    int j;
    int x;
};
struct Sparse {
    int m;
    int n;
    int num;
    struct Element *e;
};
```

```
void create (struct Sparse *s){
    printf("Enter Dimensions");
    scanf("%d%d", &s->m, &s->n);
    printf("Enter no. of non-zero");
    scanf("%d", &s->num);
    s->e = new Elements [s->num]; for (i+
    printf("Enter all elements");
    for(i=0; i<s->num; i++){
        scanf("%d%d%d", &s->e[i].i,
                &s->e[i].j, &s->e[j].x);
}
```

163. Program for adding sparse matrix

S_1 =

	1	2	3	4	5
1	0	0	3	0	0
2	4	0	0	0	7
3	0	0	5	0	8
4	0	6	0	0	0

S_2 =

	1	2	3	4	5
1	0	0	0	0	2
2	0	5	0	0	6
3	4	0	8	0	0
4	0	0	0	0	9

S_1

m	4
n	5
num	6
e	→

	0	1	2	3	4	5
i	1	2	2	3	3	4
j	3	1	5	3	5	2
x	3	4	7	5	8	6

i

S_2

m	4
n	5
num	6
e	→

	0	1	2	3	4	5
i	1	2	2	3	3	4
j	5	2	5	1	3	5
x	2	5	6	4	8	9

j

sum

m	4
n	5
num	
e	→

	1	1	2	2	2			
	3	5	1	2	5			
	3	2	4	5	13			

k

```
struct Sparse * add ( struct sparse *s1,
                      struct sparse * s2) {
  struct sparse *sum;
  if(s1->m != s2->m && s1->n != s2->n)
      return 0;
  sum = new Sparse;
  sum->m = s1->m;
  sum->n = s1->n;
  sum->e = new Element [s1->num + s2->num];
  while(i<s1->num && j<s2->num) {
    if (s1->e[i].i < s2->e[j].i)
        sum->e[k++] = s1->e[i++];
    else if (s1->e[i].i > s2->e[j].i)
        sum->e[k++] = s2->e[j++];
    else {
      if (s1->e[i].j < s2->e[j].j)
          sum->e[k++] = s1->e[i++];
      else if (s1->e[i].j > s2->e[j].j)
          sum->e[k++] = s2->e[j++];
      else {
          sum->e[k] = s1->e[i];
          sum->e[k++].x = s1->e[i++].x
                        + s2->e[j++].x;
      }
    }
  }
}
```

167. Polynomial Representation (List of terms)

$p(x) = 3x^5 + 2x^4 + 5x^2 + 2x + 7$

(exponent: 5; coefficient: 3)

$n = 5$

Coeff	3	2	5	2	7
Expo	5	4	2	1	0

```
struct Term {
    int Coeff;
    int Exp;
};
struct Poly {
    int n;
    struct Term * t;
};
```

p

n → 5

t → Coeff / Exp

	0	1	2	3	4
Coeff	3	2	5	2	7
Exp	5	4	2	1	0

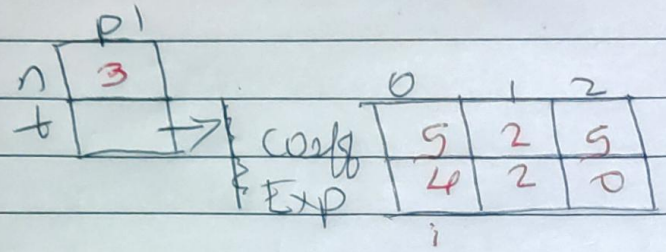
```
struct Poly p;
printf("No. of non-zero terms \n");
scanf("%d", &p.n);
p.t = new Term[p.n];        (C++)
printf("Enter Polynomial terms \n");
for(i=0; i<p.n; i++) {
    printf("Term no. %d", i+1);
    scanf("%d %d", &p.t[i].coeff,
                   &p.t[i].Exp);
}
```

168. Polynomial Evaluation

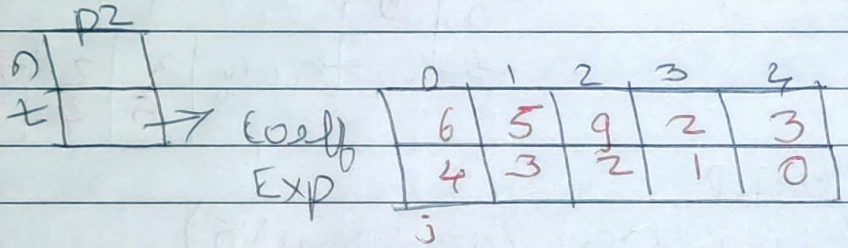
```
x=5; sum=0;
for(i=0; i<p.n; i++) {
    sum += p.t[i].coff * pow(x, p.t[i].Exp);
}
return sum;
```

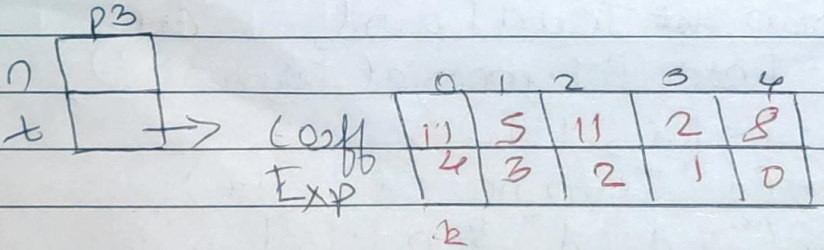
169. Polynomial Addition

$p_1(x) = 5x^4 + 2x^2 + 5$



$p_2(x) = 6x^4 + 5x^3 + 9x^2 + 2x + 3$





```
i=0; j=0; k=0;
while (i<p1.n && j<p2.n) {
    if (p1.t[i].Exp > p2.t[j].Exp)
        p3.t[k++] = p1.t[i++];
    else if (p2.t[j].Exp > p1.t[i].Exp)
        p3.t[k++] = p2.t[j++];
    else {
        p3.t[k].Exp = p1.t[i].Exp;
        p3.t[k++].coeff = p1.t[i++].coeff +
                          p2.t[j++].coeff;
    }
}
```