Section : 10

Sparse Matrix
and
Polynomial Representation

Sparse Matrix Representation -

$$
\begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\
2 & 0 & 0 & 8 & 0 & 0 & 10 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & 9 & 0 & 0 & 5 & 0 & 0 & 0 & 0
\end{array}
$$

8x9 (72 element)

72x2=144 bytes

Sparse Matrix → In numerical ~~and~~ analysis and scientific computing, a sparse matrix or sparse Array is a matrix in which most of the element are zero.

Method to store non-zero elements -

1. coordinate list (3 column representation

2. Compared Sparse row

3-Column Representation :-

Row numbes, column numbes, value (Element)

No. of rows, no. of column →

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 Sparse rows :-

A [3, 8, 10, 4, 2, 6, 9, 5] (order wise)

~~IA [0, 1, 3, 3, 7, 7, 9,~~

~~0 1 2 3 4 5 6~~

~~2+1 3+0 3+4 7+0~~

~~JA [8, 3, 6, 1, 3, 4, 2, 5]~~

IA [0, 1, 3, 3, 4, 4, 5, 6, 8]

0 1 2 3 4 5 6 7 8

↑

prev + no of element sum

IA [8, 3, 6, 1, 3, 4, 2, 5]

Space →

⇒ 8 + 9 + 8

⇒ 25 × 2

= 50 bytes

Addition of Sparse Matrices

$$A = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 6 & 0 & 0 \\ 2 & 0 & 7 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & 5 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 4 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$B = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 & 5 & 0 \\ 3 & 0 & 0 & 2 & 0 & 0 & 7 \\ 4 & 0 & 0 & 0 & 9 & 0 & 0 \\ 5 & 8 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$A + B \rightarrow C = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 6 & 0 & 0 \\ 2 & 0 & 10 & 0 & 0 & 5 & 0 \\ 3 & 0 & 2 & 2 & 5 & 0 & 7 \\ 4 & 0 & 0 & 0 & 9 & 0 & 0 \\ 5 & 12 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Non-zero element → 9

Adding using Coordinate

i i i

A

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

j j j j j

B

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

rows[i]<row[j]

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
	6	10	5	2	2	5	7	9	12

rows[i] < rows[j]

copy[i] to c and i++;

column[i]<column[j]

copy[i] to c and i++;

rows[i]==rows[j] && column[i]==column[j]

Add both element

```
rows[i] > row[j]
    copy[j] to c and j++;

column[i] > column[j]
    copy [j] to c and j++;


rows[i] == rows[j] && column[i] < column[j]
    copy[i] to c and i++;

rows[i] == rows[j] && column[i] > column[j]
    copy[j] to c and j++;
```

Array Representation of Sparse Matrix :-

$$A = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 0 & 7 & 0 & 0 \\ 2 & 2 & 0 & 0 & 5 & 0 \\ 3 & 9 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 4 \end{array}$$

$\underline{4 \times 5}$

Non-zero element = 5

dimension and nozero element

	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

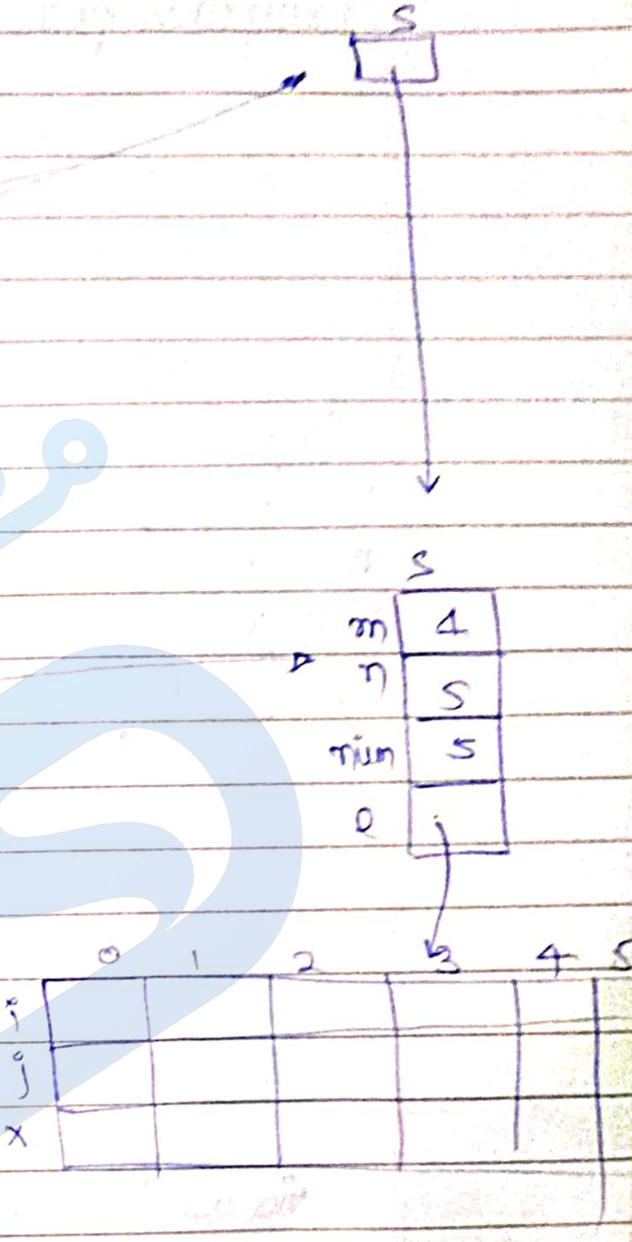
$\underline{m \times n}$

- How to represent
- How to create
- How to add

```
Struct Element
{
        int i;
        int j;
        int x;

};
```

```
struct sparse
{
      int m;
      int n;
      int num;
      struct Element *e;   // Dynamic Array
                  ↓
   }
              Array of element


   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 Element[s->num];

            printf("Enter all elements");

            for (int i=0; i<num; i++)
            {

                        scanf("%d%d%d", &s->e[i].i,

                                      &s->e[j].j,

                                      &s->e[i].x);

            }
   }
```

```
void main()
{
    struct sparse s;

     create(&s);

}
```



Program for Adding Sparse Matrix →

$$S_1 = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 0 & 3 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 7 \\ 3 & 0 & 0 & 5 & 0 & 8 \\ 4 & 0 & 6 & 0 & 0 & 0 \end{array} \qquad S_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 5 & 0 & 0 & 6 \\ 4 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}$$

S1

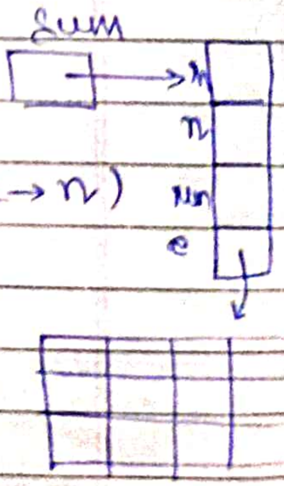
m	4
n	5
num	6
e	→

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

S2

m	4
n	5
num	6
e	↓

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



```
void add(struct sparse *s1, struct sparse *s2)
{
        if (s1->m != s2->m || s1->n != s2->n)
        {
              return 0;
        }
        sparse *sum;

        sum = new sparse;

        sum->m = s1->m ; sum->n = s2->n;
```

```
sum→e = new Element [s1→num + s2→num];

while( i< s1→ num && j< s2→num)
{   ~~if(s1→e[i].i < s2→~~
    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++] ;

              }
              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+= s2→e[j++].x;
                 }
}             }
```

Pg. No

Date / /

Polynomial Representation

$$p(x) = \overset{\text{coefficient}}{3}\underset{\text{variable}}{x}^{\overset{\text{exponent}}{5}} + 2x^4 + 5x^2 + 2x + 7$$

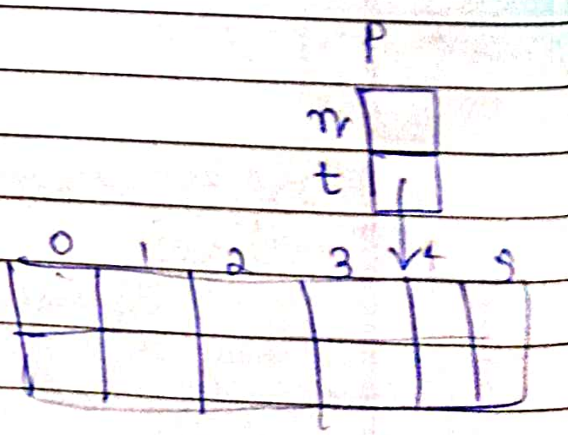
1) Polynomial Representation
2) Evaluation of Polynomial
3) Addition of two polynomials

n=5

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

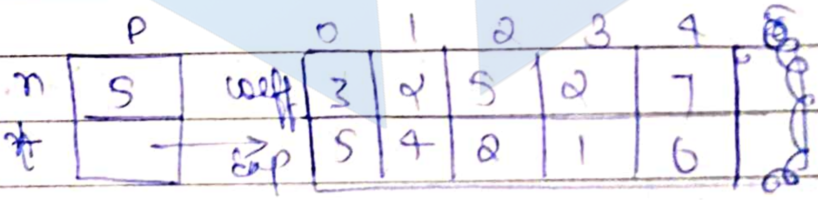
```
struct Term
{
    int coeff;
    int Expo;
};
```

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



```
struct Poly P;
printf(" No. of non-zero element terms");
 scanf("%d", & p.n);

  p.t= new term[P.n]

   printf("Enter Polynomial terms");
    for (i=0; i<p.n; i++)
    {
          printf("Term no %d", i+1)

          scanf("%d %d", &p.t[i].coeff,
                            &p.t[i].Expo);
     }
```

Polynomial evaluation →

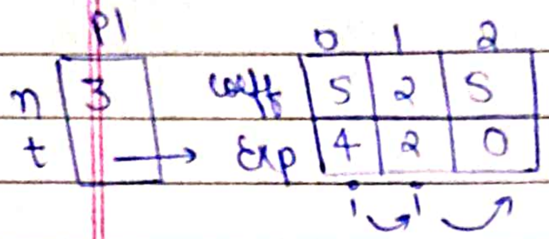


	0	1	2	3	4
coeff	3	2	5	2	7
exp	5	4	2	1	6

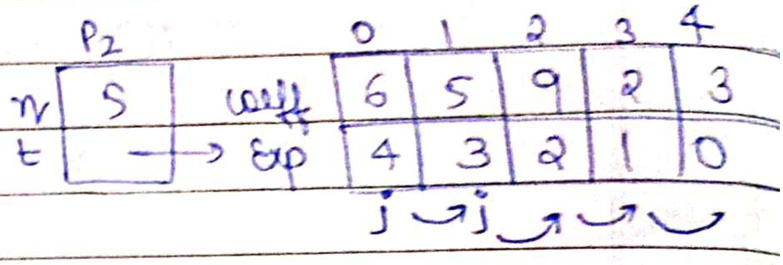
```
struct Poly P;
  int sum=0;
 for(i=0; i<p.n; i++)
  {
 sum+=p.t[i].coeff * Pow(x, p.t[i].exp);

  }
   return sum;
```

Polynomial Addition:-

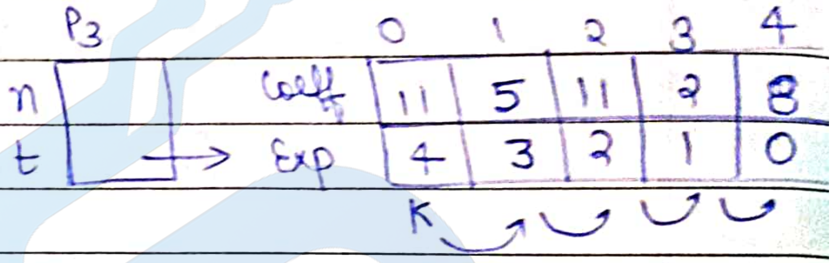
$P1(x) = 5x^4 + 2x^2 + 5$ $\qquad$ $P2(x) = 6x^4 + 5x^3 + 9x^2 + 2x + 3$



	0	1	2
coeff	5	2	5
Exp	4	2	0



	0	1	2	3	4
coeff	6	5	9	2	3
Exp	4	3	2	1	0



	0	1	2	3	4
coeff	11	5	11	2	8
Exp	4	3	2	1	0

```
i=j=K=0;

while(i<p1.n && j<p2.n)
{

        if( p1.t[i].Exp >P2. t[j].Exp)
        {

                P3.t[K++]=P1.t[i++]};

        }
        elseif( 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;

    }

}
```