Unit II

2 Linear Data Structure using Sequential Organization

2.1 : Concept of Sequential Organization

Q.1 What is arrays ?

Ans. : Defintion of Arrays : Array is a set of consecutive memory locations which contains similar data elements.

Array is basically a **set of pair-index and the value.**

Syntax

```
data_type  name_of_array [size] ;
```

For example, int a [10]; double b[10] [10];

Here 'a' is the name of the array inside the square bracket size of the array is given. This array is of integer type i.e. all the elements are of integer type in array 'a'.

Q.2 Give the advantages and disadvantages of sequential organization of data structure.

Ans. : Advantages of sequential organization of data structure

1. Elements can be retrieved or stored very efficiently in sequential organization with the help of index or memory location.
2. All the elements are stored at continuous memory locations. Hence searching of element from sequential organization is easy.

Disadvantages of sequential organization of data structure

1. Insertion and deletion of elements becomes complicated due to sequential nature.
2. For storing the data large continuous free block of memory is required.
3. Memory fragmentation occurs if we remove the elements randomly.

2.2 : Array as an Abstract Data Type

Q.3 Write an ADT for Arrays.

Ans. :

AbstractDataType Array

{

Instances : An array A of some size, index i and total number of elements in the array n.

Operations :

1. **Create** () – This operation creates an array.
2. **Insert**() - This operation is for inserting the element in an array
3. **Delete**() - This operation is for deleting the elements from the array. Only logical deletion of the element is possible.
4. **display** () – This operation displays the elements of the array.

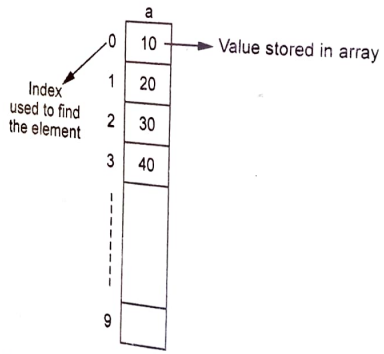
}

2.3 : Array Overview

Q.4 Explain the types of arrays.

Ans. : One dimensional array :

The one dimensional array 'a' is declared as int a[10];



Two dimensional array :

If we declare a two dimensional array as

int a[10] [3];

Then it will look like this -

Column ↓

row →		0	1	2
	0	10	20	30
	1	40	50	60
	2			
	.			
	.			
	.			
	9			

The two dimensional array should be in row-column form.

2.4 : Operations on Array

Q.5 Write a Python program to insert an element in an array.

Ans. :

Python Program

```
print("\nHow many elements are there in Array?")
n = int(input())
array = []
i=0
for i in range(n):
    print("\n Enter element in Array")
    item = int(input())
     array.append(item)
print("Enter the location where you want to insert an element")
position = int(input())

print("Enter the value to insert")
value = int(input())
array=array[:position]+[value]+array[position:]
```

```
print("Resultant array is\n")
print(array)
```

Q.6 Explain the traversal of an array.

Ans. :

- The loop is used in list for traversing purpose. The for loop is used to traverse the list elements.

Syntax

```
for VARIABLE in LIST :
    BODY
```

Example

```
>>> a=['a','b','c','d','e'] # List a is created
>>> for i in a:
print(i)
will result into
a
b
c
d
e
>>>
```

Q.7 Write a Python program to delete an element from an array.

Ans. : Python Program

```
print("\nHow many elements are there in Array?")
n = int(input())
array = []
i=0
for i in range(n):
    print("\n Enter element in Array")
    item = int(input())
    array.append(item)
print("Enter the index from where you want to delete an element")
position = int(input())
array=array[:position]+array[position+1:]
```


print("Resultant array is\n")
print(array)

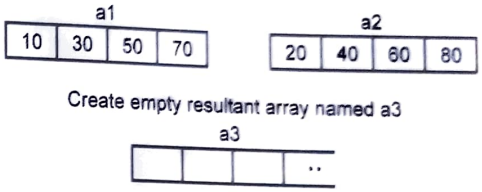
2.5 : Merging of Two Arrays

Q.8 How to merge two arrays? Explain it with suitable example.

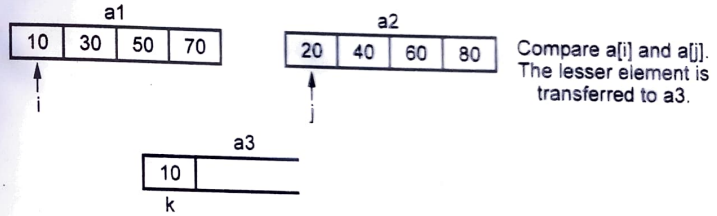
Ans : Merging of two arrays result into a single array in which elements are arranged in sorted order.

For example - Consider two arrays as follows

Step 1 :

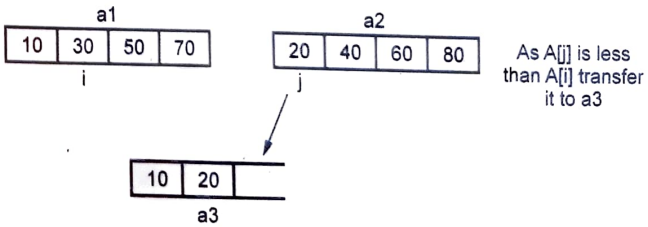


Step 2 :



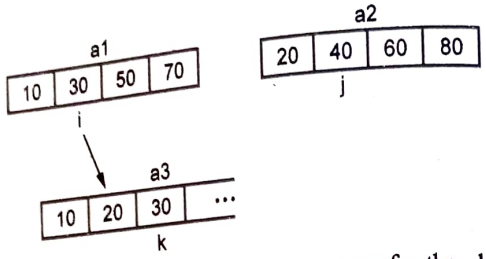
As a1[i] < a2[j], transfer element a1[i]to a3[k]. Then increment i and k pointer

Step 3 :



As a2[j] is transferred to a3 array increment j and k pointer.

Step 4 :



Increment i and k pointers. In this way we can transfer the elements from two arrays a1 and a2 to get merged array a3.

Finally we get

10	20	30	40	50	60	70	80

Q.9 Write a Python function for merging of two arrays.

Ans. : Python Program

```
def mergeArr(a1,a2,n,m):
    a3 = [None]*(n+m)
    i = 0
    j = 0
    k = 0
    #traverse both arrays
    #if element of first array is less then store it in third array
    #if element of second array is less then store it in third array
    while i < n and j < m:
        if a1[i] < a2[j]:
            a3[k] = a1[i]
            k = k + 1
            i = i + 1
        else:
            a3[k] = a2[j]
            k = k + 1
            j = j + 1
    #if elements of first array are remaining
    #then transfer them to third array
    while i < n:
        a3[k] = a1[i]
```



```
        k = k + 1
        i = i + 1

    #if elements of second array are remaining
    #then transfer them to third array
    while j < m:
        a3[k] = a2[j]
        k = k + 1
        j = j + 1

    #display the resultant merged array
    print("Merged Array is ...")
    for i in range (n+m):
        print(str(a3[i]), end = " ")
```

2.6 : Storage Representation and their Address Calculation

Q.10 Explain two dimensional arrays with row and column major implementation. Explain address calculation in both cases with example ☞ **[SPPU : Dec.-18, Marks 6]**

Ans. : Row Major Representation

If the elements are stored in **row wise manner** then it is called row major representation.

For example : If we want to store elements

10 20 30 40 50 60 then in a two dimensional array

The ⇒ elements will be stored horizontally

	0	1	2
0	10	20	30
1	40	50	60
.			
.			
.			
9			

Column Major Representation

If elements are stored in **column wise manner** then it is called column major representation.

For example : If we want to store elements

10 20 30 40 50 60 then the elements will be filled up by column wise manner as follows (consider array a[3] [2]). Here 3 represents number of rows and 2 represents number of columns.

	0	1
0	10	40
1	20	50
2	30	60

Example : Refer Q.11.

Q.11 Consider integer array int arr[4][5] declared in 'C' program. If the base address is 1020, find the address of the element arr[3][4] with row major and column major representation of array.

☞ **[SPPU : Dec.-09, Marks 6]**

Ans. : Row Major Representation

The element a[i][j] will be at

a[i][j] = base address+(row_index*total number of columns + col_index)*element_size

= (base address+(i*col_size+j)*element_size)

when i=3 and j=4 ,element_size=int occupies 2 bytes of memory hence it is 2

total number of columns=5

a[3][4] = 1020+(3*5+4)*2

= 1020+38

a[3][4] = 1058

Column Major Representation

The element a[i][j] will be at

a[i][j] = base address+(col_index*total number of rows+row_index)*element_size

= (base address+(j*row_size+i)*element_size)

when i=3 and j=4, element_size=int occupies 2 bytes of memory hence it is 2

total number of rows = 4

a[3][4] = 1020+(4*4+3)*2

= 1020+38

a[3][4] = 1058

2.7 : Multidimensional Arrays

Q.12 Write a python program for performing addition of two matrices.

Ans. :

```
def add_matrix(arr1,arr2):
    result = [[arr1[i][j] + arr2[i][j]  for j in range(len(arr1[0]))] for i in
             range(len(arr1))]

    print("The Addition of Two Matrices...")
    print(result)
row_num = int(input("Input number of rows: "))
col_num = int(input("Input number of columns: "))
arr1 = [[0 for col in range(col_num)] for row in range(row_num)]
for row in range(row_num):
    for col in range(col_num):
        item = int(input("Enter the elements in first matrix: "))
        arr1[row][col]= item
print("The first matrix is...")
print(arr1)

arr2 = [[0 for col in range(col_num)] for row in range(row_num)]
```

```
for row in range(row_num):
    for col in range(col_num):
        item = int(input("Enter the elements in second matrix: "))
        arr2[row][col]= item

print("The second matrix is...")
print(arr2)

#Driver Code
add_matrix(arr1,arr2,)
```

Output

```
Input number of rows: 3
Input number of columns: 3
Enter the elements in first matrix: 1
Enter the elements in first matrix: 2
Enter the elements in first matrix: 3
Enter the elements in first matrix: 4
Enter the elements in first matrix: 5
Enter the elements in first matrix: 6
Enter the elements in first matrix: 7
Enter the elements in first matrix: 8
Enter the elements in first matrix: 9
The first matrix is...
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Enter the elements in second matrix: 1
Enter the elements in second matrix: 1
Enter the elements in second matrix: 1
Enter the elements in second matrix: 2
Enter the elements in second matrix: 2
Enter the elements in second matrix: 2
Enter the elements in second matrix: 3
```



```
Enter the elements in second matrix: 3
Enter the elements in second matrix: 3
The second matrix is...
[[1, 1, 1], [2, 2, 2], [3, 3, 3]]
The Addition of Two Matrices...
[[2, 3, 4], [6, 7, 8], [10, 11, 12]]
>>>
```

(2) Matrix Multiplication

Consider matrix A

1	2	3
4	5	6
7	8	9

Consider matrix B

1	1	1
2	2	2
3	3	4

The resultant matrix is

14	14	17
32	32	38
50	50	59

Q.13 Write a python program to implement matrix multiplication operation.

Ans. :

```
A = [[1,2,3],
     [4,5,6],
     [7,8,9]]

B = [[1,1,1],
     [2,2,2],
     [3,3,4]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]
```

```
print("Matrix A is ...")
print(A)

print("Matrix B is ...")
print(B)

# iterate through rows of X
for i in range(len(A)):
   for j in range(len(B[0])):
      for k in range(len(B)):
         result[i][j] += A[i][k] * B[k][j]

print("Matrix Multiplication is ...")
for r in result:
   print(r)
```

Output

```
Matrix A is ...
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Matrix B is ...
[[1, 1, 1], [2, 2, 2], [3, 3, 4]]
Matrix Multiplication is ...
[14, 14, 17]
[32, 32, 38]
[50, 50, 59]
>>>
```

(3) Transpose of Matrix

Consider matrix A

1	2	3
4	5	6
7	8	9

Transposed matrix

1	4	7
2	5	8
3	6	9

Q.14 Write a python program for performing transpose of matrix.

Ans. :

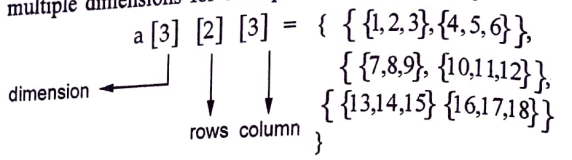
```
A = [[1,2,3],
     [4,5,6],
     [7,8,9]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

print("Original Matrix is...")
print(A)

# iterate through rows
for i in range(len(A)):
   for j in range(len(A[0])):
      result[j][i] = A[i][j]

print("Transposed Matrix is ...")
for r in result:
   print(r)
```

Output

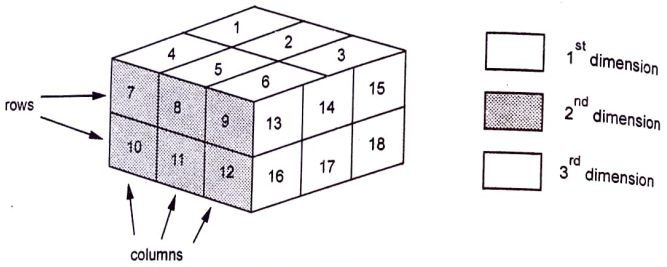
```
Original Matrix is...
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Transposed Matrix is ...
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]
>>>
```

Q.15 Write a short note on - Three dimensional arrays.

Ans. : The multidimensional array is similar to the two dimensional array with multiple dimensions for example Here is 3-D array



We can represent it graphically as



2.8 : Concept of Ordered List

Q.16 Explain the concept of ordered list.

Ans. : Ordered list is nothing but a set of elements. Such a list sometimes called as linear list.

For example

1. List of one digit numbers

   (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

2. Days in a week.

   (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday)

With this concept in mind let us formally define the ordered list.

Definition : An ordered list is set of elements where set may be empty or it can be written as a collection of elements such as (a1, a2, a3 ..... an).

Q.17 Write a python program to create a list of even numbers from 0 to 10.

Ans. :

```
even = [] #creating empty list
for i in range(11):
   if i % 2 ==0:
      even.append(i)
print("Even Numbers List: ",even)
      even = [] #creating empty list
for i in range(11):
   if i % 2 ==0:
      even.append(i)
print("Even Numbers List: ",even)
```

Output

```
Even Numbers List: [0, 2, 4, 6, 8, 10]
```

Q.18 Write a python program to combine and print two lists using list comprehension.

Ans. :

```
print([(x,y)for x in['a','b'] for y in ['b','d'] if x!=y])
```

Output

```
[('a', 'b'), ('a', 'd'), ('b', 'd')]
```

2.9 : Single Variable Polynomial

Q.19 Explain polynomial representation using arrays.

☞[SPPU : May 17, 18, Marks 3]

Ans. : Definition : Polynomial is the sum of terms where each term consists of variable, **coefficient** and **exponent.**

Representation : For representing a single variable polynomial one can make use of one dimensional array. In single dimensional array the index of an array will act as the exponent and the coefficient can be stored at that particular index which can be represented as follows :

For e.g. : $3x^4 + 5x^3 + 7x^2 + 10x - 19$

This polynomial can be stored in single dimensional array.

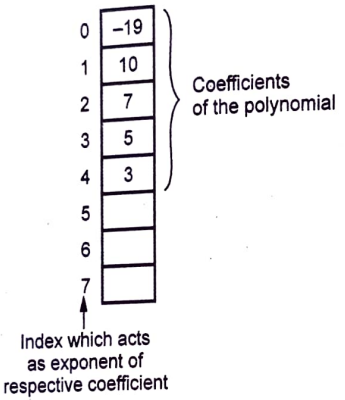


Fig. Q.19.1 Polynomial representation

Q.20 Write a program to add two polynomials.

Ans. :

```
#include<iostream>
using namespace std;
class APADD
{
   private:
      struct p
      {
         int coeff;
         int expo;
      };
   public:
      p p1[10],p2[10],p3[10];
      int Read(p p1[10]);
      int add(p p1[10],p p2[10],int t1,int t2,p p3[10]);
      void Print(p p2[10],int t2);
};
```



```
/*
-------------------------------------------------------------------
The Read Function Is For Reading The Two Polynomials
-------------------------------------------------------------------
/*
int APADD::Read(p p[10])
{
   int t1,i;
   cout<<"\n Enter The Total number Of Terms in The Polynomial: ";
   cin>>t1;
   cout<<"\n Enter The Coef and Exponent In Descending Order";
   for(i=0;i<t1;i++)
   {
      cout<<"\n Enter Coefficient and exponent: ";
      cin>>p[i].coeff;
      cin>>p[i].expo;
   }
   return(t1);
}

/*
-------------------------------------------------------------------
The add Function Is For adding The Two Polynomials
-------------------------------------------------------------------*/
int APADD::add(p p1[10],p p2[10],int t1,int t2,p p3[10])
{
   int i,j,k;
   int t3;
   i=0;
   j=0;
   k=0;
   while(i<t1 &&j<t2)
   {
      if(p1[i].expo==p2[j].expo)
      {
```

```
            p3[k].coeff=p1[i].coeff+p2[j].coeff;
            p3[k].expo =p1[i].expo;
            i++;j++;k++;
        }
        else if(p1[i].expo>p2[j].expo)
        {
            p3[k].coeff=p1[i].coeff;
            p3[k].expo =p1[i].expo;
            i++;k++;
        }
        else
        {
            p3[k].coeff=p2[j].coeff;
            p3[k].expo =p2[j].expo;
            j++;k++;
        }
    }
    while(i<t1)
    {
        p3[k].coeff=p1[i].coeff;
        p3[k].expo =p1[i].expo;
        i++;k++;
    }
    while(j<t2)
    {
        p3[k].coeff=p2[j].coeff;
        p3[k].expo =p2[j].expo;
        j++;k++;
    }
    t3=k;
    return(t3);
}
/*
-----------------------------------------------------------
The Print Function Is For Printing The Two Polynomials
-----------------------------------------------------------
```



```
*/
void APADD::Print(p pp[10],int term)
{
    int k;
    cout<<"\n Printing The Polynomial";
    for(k=0;k<term-1;k++)
        cout<<" "<<pp[k].coeff<<"X^"<<pp[k].expo<<"+ ";
    cout<<pp[k].coeff<<"X^"<<pp[k].expo;
}
/*
-----------------------------------------------------------
 The main function
-----------------------------------------------------------
*/
int main()
{
    APADD obj;
    int t1,t2,t3;
    cout<<"\n Enter The First Polynomial";
    t1=obj.Read(obj.p1);
    cout<<"\n The First Polynomial is: ";
    obj.Print(obj.p1,t1);
    cout<<"\n Enter The Second Polynomial";
    t2=obj.Read(obj.p2);
    cout<<"\n The Second Polynomial is: ";
    obj.Print(obj.p2,t2);
    cout<<"\n The Addition is: ";
    t3=obj.add(obj.p1,obj.p2,t1,t2,obj.p3);
    obj.Print(obj.p3,t3);
    return 0;
}
```

2.10 : Sparse Matrix

Q.21 What is sparse matrix? Explain it with example.

☞ [SPPU : May-19, Marks

Ans. : • **Definition :** Sparse matrix is a matrix containing few non ze elements.

- For example - if the matrix is of size 100 × 100 and only 10 elemen are non zero. Then for accessing these 10 elements one has to mak 10000 times scan. Also only 10 spaces will be with non-zero elemen remaining spaces of matrix will be filled with zeros only. i.e. we hav to allocate the memory of 100 × 100 × 2 = 20000.
- Hence sparse matrix representation is a kind of representation in whic only non zero elements along with their rows and columns is stored.

1	2	3
4	5	6
7	8	9

Dense Matrix

1	0	0
0	0	0
0	1	0

Sparse Matrix

Q.22 Write a function for addition of two sparse matrices.

Ans. :

```
void add(int s1[10][3],int s2[10][3],int s3[10][3])
{
   int i, j, k;
   i = 1; j = 1; k = 1;
   if ((s1[0][0] == s2[0][0]) && (s1[0][1] == s2[0][1]))
   {
      s3[0][0] = s1[0][0];
      s3[0][1] = s1[0][1];
   //traversing thru all the terms
      while ((i <= s1[0][2]) && (j <= s2[0][2]))
      {
```



```
      if (s1[i][0] == s2[j][0])
      {
         if (s1[i][1] == s2[j][1])
         {
            s3[k][2] = s1[i][2] + s2[j][2];
            s3[k][1] = s1[i][1];
            s3[k][0] = s1[i][0];
            i++;
            j++;
            k++;
         }
         else if (s1[i][1]<s2[j][1])
         {
            s3[k][2] = s1[i][2];
            s3[k][1] = s1[i][1];
            s3[k][0] = s1[i][0];
            i++;
            k++;
         }
         else
         {
            s3[k][2] = s2[j][2];
            s3[k][1] = s2[j][1];
            s3[k][0] = s2[j][0];
            j++;
            k++;
         }
      }//end of 0 if
      else if (s1[i][0]<s2[j][0])
      {
         s3[k][2] = s1[i][2];
         s3[k][1] = s1[i][1];
         s3[k][0] = s1[i][0];
         i++;
         k++;
```

```
        }
        else
        {
            s3[k][2] = s2[j][2];
            s3[k][1] = s2[j][1];
            s3[k][0] = s2[j][0];
            j++;
            k++;
        }
    }//end of while
    //copying remaining terms
    while (i <= s1[0][2])
    {
        s3[k][2] = s1[i][2];
        s3[k][1] = s1[i][1];
        s3[k][0] = s1[i][0];
        i++;
        k++;
    }
    while (j <= s2[0][2])
    {
        s3[k][2] = s2[j][2];
        s3[k][1] = s2[j][1];
        s3[k][0] = s2[j][0];
        j++;
        k++;
    }
    s3[0][2] = k - 1;
}
else
    cout<<"\n Addition is not possible";
}
```

Q.23 Explain fast transpose of sparse matrix with suitable example. Discuss time complexity of fast transpose.

[SPPU : May-17, Marks

Ans. : The fast transpose is a transpose method in which matrix transpose operation is performed efficiently. In this method an auxiliary array are used to locate the position of the elements to be transposed sequentially.

Logic for Fast Transpose

Consider the sparse matrix representative as

S1

Index	Row	Col	Non-Zero values
0	3	3	4
1	0	1	10
2	1	0	20
3	2	0	30
4	2	1	40

We will first consider one dimensional array, named **rterm[]**. In this array we will store non zero terms present in each column.

At 0^{th} column there are two non zero terms. At 1^{st} column also there are two non zero terms but there is non zero term in 2^{nd} column.

	rterm
0	2
1	2
2	0

Similarly we will take another one dimensional array named **rpos[]**. We initialize 0^{th} location of **rpos[]** by 1. so,

	rpos
0	1

Now we use following formula to fill up the **rpos** array.

$$rpos[i] = rpos[i - 1] + rterm[i - 1]$$

i = 1

rpos[1] = rpos[0] + rterm[0]

= 1 + 2

rpos[1] = 3

i = 2

rpos[2] = rpos[1] + rterm[1]

= 3 + 2

rpos[2] = 5

i = 3

rpos[3] = rpos[2] + rterm[2]

= 5 + 0 = 5

rpos[3] = 5

rpos

0	1
1	3
2	5
3	5

Now we will read values of S1 array from 1 to 4.

We must find the triplet from S1 array which is at index 1. It is (0, 1, 10)

index	row	col	value
1	0	1	10

S1

Just interchange row and col.

row	col	value
1	0	10

Since value is 1, check rpos[1]

rpos[1] points to value 3. That means place the triplet (1, 0, 10) at index 3 in S2 array.

S2

	Row	Col	Non-Zero values
0			
1			
2			
3	1	0	10
4			

Now since rpos[1] is read just now, increment rpos[1] value by 1.

Hence

rpos

0	1
1	~~3~~ 4
2	5
3	5

Read next element from S1 array

index	row	col	value
2	1	0	20

Interchange →

row	col	value
0	1	20

rpos[0] = 1

That mean place triplet (0, 1, 20) at index 1 in S2 array

S2

	Row	Col	Non-Zero values
0			
1	0	1	20
2			
3	1	0	10
4			

Since rpos[0] is read just now, increment rpos[0] by 1.

rpos

0	~~1~~ 2
1	4
2	5
3	5

Read next element from S1 array

index	row	col	value
3	2	0	30

Interchange

row	col	value
0	2	30

rpos[0] = 2

That means place triplet (0, 2, 30) at index 2 in S2 array.

S2

	Row	Col	Non-Zero values
0			
1	0	1	20
2	0	2	30
3	1	0	10
4			

Since rpos[0] is read just now increment rpos[0] by 1.

	rpos	
0	2	3
1	4	
2	5	
3	5	

Read next element from S1 array

index	row	col	value
4	2	1	40

Interchange

row	col	value
1	2	40

rpos[1] = 4

That means place triple (1, 2, 40) at index 2 in S2 array.

S2

	Row	Col	Non-Zero values
0			
1	0	1	20
2	0	2	30
3	1	0	10
4	1	2	40

Thus we get transposed sparse matrix.

Finally we fill up S2[0] by total number of rows, total number of colums and total number of non-zero values.

Thus we get

	Row	Col	Non-Zero values
0	3	3	4
1	0	1	20
2	0	2	30
3	1	0	10
4	1	2	40

C++ Code

```
void trans (int s1[max1][3],int s2[max1][3] )
{
    int rterm[max1],rpos[max1];
    int j,i ;
    int row,col,num ;

    row= s1[0][0];
    col= s1[0][1];
    num= s1[0][2];
```

```
    s2[0][0] = col;
    s2[0][1] = row;
    s2[0][2] = num;
    if ( num > 0 )
    {
        for ( i = 0; i <= col ; i ++ )
            rterm[i] = 0;
        for ( i = 1; i <= num ; i ++ )
            rterm[s1[i][1]] ++;
        rpos[0] = 1;  /*setting the rowwise position*/
        for ( i = 1; i <= col; i++ )
         rpos[i]=rpos[i-1]+ rterm[(i - 1)];
        for ( i = 1; i <= num ; i ++ )
        {
            j = rpos[s1[i][1]];
            s2[j][0] = s1[i][1];
            s2[j][1] = s1[i][0];
            s2[j][2] = s1[i][2];
            rpos[s1[i][1]] = j + 1;
        }
    }
}
```

Time Complexity of Fast Transpose

For transposing the elements using simple transpose method we need two nested for loops but in case of fast transpose we are determining the position of the elements that get transposed using only one for loop. Hence the time complexity of fast transpose is O(n)

2.11 : Time and Space Tradeoff

Q.24 Explain the concept of time and space tradeoff.

Ans. : Basic concept : Time space trade-off is basically a situation where either a space efficiency (memory utilization) can be achieved at

the cost of time or a time efficiency (performance efficiency) can be achieved at the cost of memory.

Example : Consider the programs like compilers in which symbol table is used to handle the variables and constants. Now if entire symbol table is stored in the program then the time required for searching or storing the variable in the symbol table will be reduced but memory requirement will be more. On the other hand, if we do not store the symbol table in the program and simply compute the table entries then memory will be reduced but the processing time will be more.

END...