

ASSIGNMENT

ASSIGNMENT-NO - 1

SUBJECT - DATA STRUCTURE USING
C & C++

UNIT-I

Date _____
Page _____

(Assignment - 1) Data Structure using C++

Que-1 What is a data structure?
Explain its characteristics with suitable real-life examples.

Ans - A data structure is a way of organizing and storing data in a computer so that it can be accessed and modified efficiently. It defines the relationship between the data and the operation that can be performed on the data.

Characteristics of Data Structures

Connectness

Time complexity

Space complexity

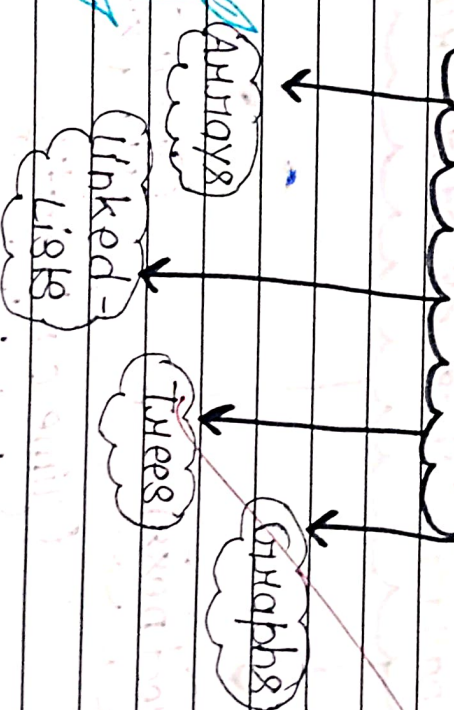
⑨

connectness - The data structure should correctly implement its interface

Time complexity - The efficiency of operations (insertion, deletion, searching etc)

Space complexity - The amount of memory space required to store the data and perform operation

Real-life examples



3

Анкора -: A list of items in a shopping cart.

Linked lists -: A chain of instructions in a treasure hunt.

Треш -: A family tree showing hierarchical relationships

Графы -: A map showing connections between cities

Q

[Question-No-2]

Write short notes on the representation of single dimensional and multidimensional arrays in memory with an example

(Answer)

Single-Dimensional arrays

A single-dimensional array is a collection of elements of the same data type stored in contiguous memory location. Each element is identified by its index or subscript.

5

- Example -

```
#include <iostream>
```

```
using namespace
```

```
int main() {
```

```
    int arr[5] = { 10, 20, 30, 40,
```

```
                50};
```

```
    for (int i = 0; i < 5; i++) {
```

```
        std::cout << "Element at index "
```

```
        << i << " : " << arr[i]
```

```
        << " | Address : " << &arr[i]
```

```
        << " ) " << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

2

Date _____
Page _____

Multidimensional Arrays :-

Multidimensional arrays use more than one subscript to describe the array's element.

2D Array :- It is use subscript one subscript to represent row value and second subscript column value

Example -

```
#include <iostream>
using namespace std;
int main() {
    int arr [2][3] = {1, 5, 4, 8, 3, 2}
    for (int i=0; i<2; i++) {
        for (int j=0; j<3; j++) {
            cout << arr [i][j];
        }
    }
```

```
} cout << end;  
}
```

```
return 0;  
}
```

[Question - 3]

What is a Sparse Array? Why is it used? Write the representation of sparse array with example

(Answer)

A sparse array is a type of array that contains a large number of zero or null elements with only a few non-zero element.

Sparse arrays are used to efficiently store and manipulate large datasets that contains many zero or null values.

8

[Example]

0	0	2	
0	0	0	
3	0	0	

few elements
are not zero

Declaration -

```
int sparse [ rows ] [ column ];
```

Initialization

```
int sparse [ 3 ] [ 3 ] = {
```

```
{ 0, 0, 2 };
```

```
{ 0, 0, 0 };
```

```
{ 0, 0, 3 };
```

```
}
```

Program -:

```
# include <iostream>
```

```
using namespace std;
```

```
using main() {
```

```
int sparse [ 3 ] [ 3 ]
```

Q

Date _____
Page _____

```
cout << " sparse matrix\n";  
{  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 3; j++) {  
            cout << sparse[i][j] << " ";  
        }  
        cout << endl;  
    }  
    return 0;  
}
```

[Question - 4]

Explain lower triangular matrix and upper triangular matrix with suitable example. How can they be stored using one-dimensional array?

(Answer)

A lower triangular matrix is a square matrix in which all elements above the diagonal are 0.

Example -: nxn matrix, vector size =

$n/(n+1)/2$;

Declaration - `int lower( $n*(n+1)/2$ );`

Initialization -

(see, for $n=3$)

matrix form

1 0 0

4 5 0

7 8 9

vector representation

Program -:

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
int n = 3;
```

```
int lower [6] = {1, 4, 5, 7, 8, 9}
```

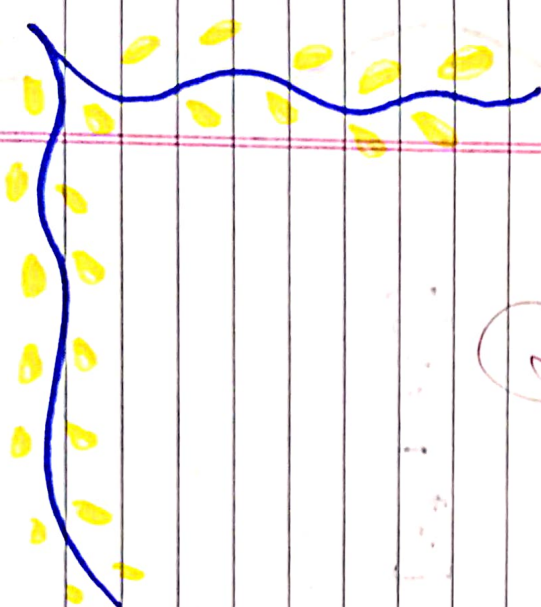
```
cout << "lower triangular matrix\n"
```

```

int index = 0;
for (int i = 0; i < n; i++) {
    if (j <= i) {
        cout << lower[index++] << " ";
    }
    else {
        cout << "0 ";
    }
    cout << endl;
}
return 0;

```

2



Question - No - 5

What is a Tri-diagonal matrix
write its vector representation
and explain with example.

(Answer)

A Tri-diagonal matrix has non
zero element only on the
main diagonal the diagonal
just above it and the
diagonal -

For $n \times n$ - matrix Vector
Size = $3n - 2$

Declaration :-

int tri [$3 * n = 2$];

Initialization :-

matrix form

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

vector representation

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Program -

```
#include <iostream>
using namespace std;
int main() {
    int n = 4;
    int tri[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    cout << "Tri diagonal matrix\n";
    int k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (j == i || j == i - 1 || j == i + 1) {
                cout << tri[k++] << " ";
            }
        }
    }
```

14

Date _____
Page _____

```
else {  
    cout << "0";  
}  
return 0;  
}
```

(Question-4)

Upper triangular matrix (vector representation -

A upper triangular matrix is square matrix in which all element below the diagonal are 0; for $n \times n$ matrix vector size = $n \cdot (n+1) / 2$

Declaration :-

```
int upper[ $n \cdot (n+1) / 2$ ];
```

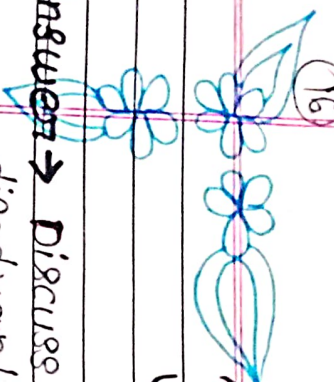
Initialization :-

Matrix form

```
1 2 3  
0 5 6  
0 0 9
```

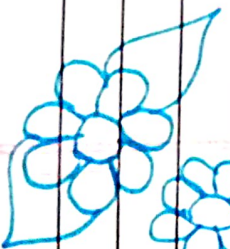
Vector representation [1, 2, 3, 5, 6, 9]

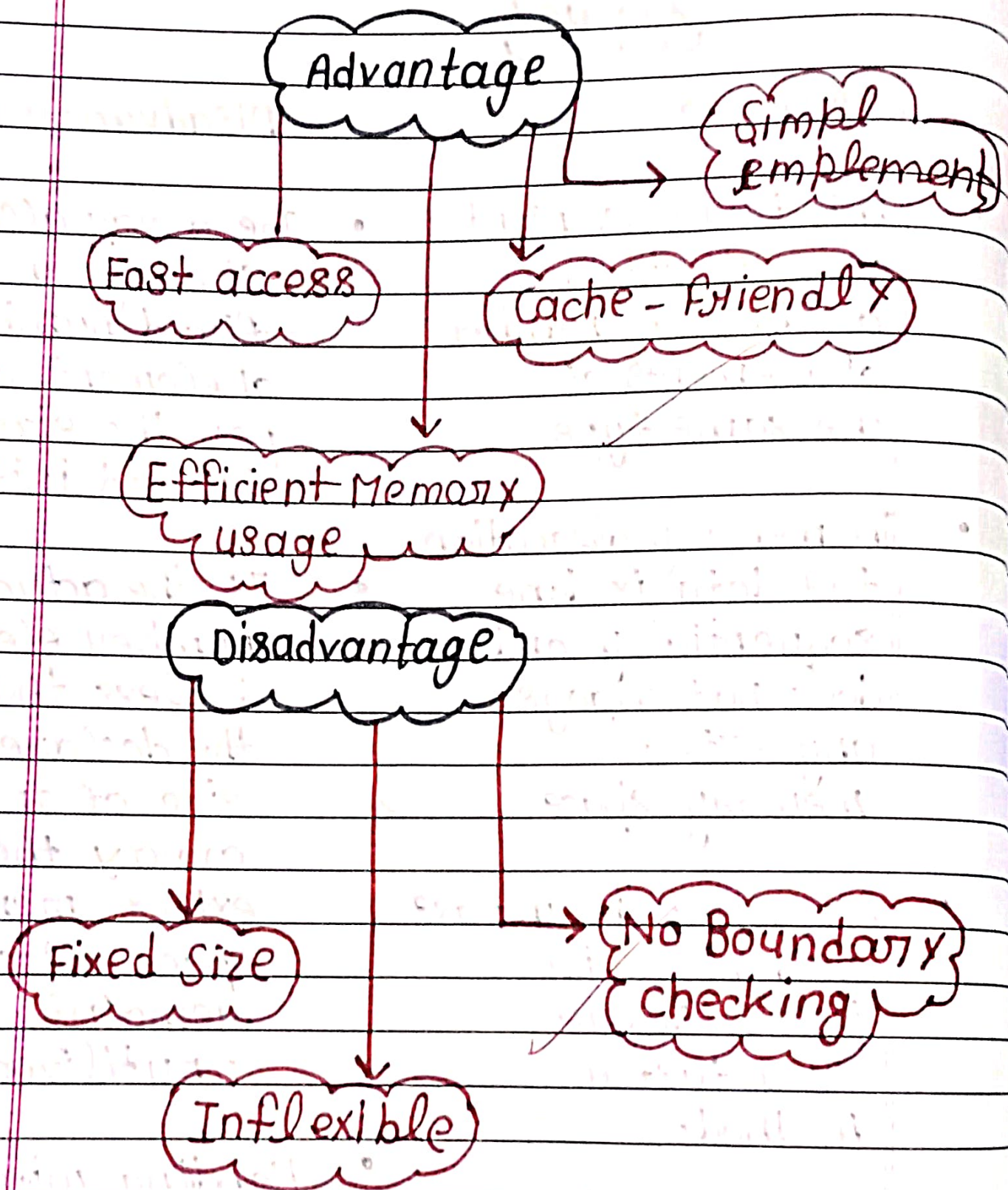
```
Program :-  
#include <iostream>  
using namespace std;  
int main() {  
    int n = 3;  
    int upper[6] = {1, 2, 3, 5, 6, 9};  
    cout << "Upper triangular Matrix is/n";  
    int index = 0;  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            if (j >= i) {  
                cout << upper[index++] << " ";  
            }  
            else {  
                cout << "0 ";  
            }  
        }  
        cout << endl;  
    }  
    return 0;  
}
```



Question - 6

Answer → Discuss the advantage and disadvantage of using arrays as a data structure
Give suitable example →





Advantage



Fast Access -: Arrays allow for fast access to elements using their index.

Efficient memory usage -: Arrays store elements in contiguous memory locations making them memory-efficient.

Cache-Friendly -: Arrays are cache-friendly which improves performance.

Simple Implementation -: Arrays are a fundamental data structure with a simple implementation.

Disadvantage

Fixed size -: Arrays have a fixed size which can lead to wasted memory or insufficient space.

20

(2) **Inflexible** -: Arrays are inflexible making insertion and deletion expensive

(3) **No boundary checking** -: Arrays do not perform boundary checking, which can lead to errors.

(11)

Answer

Advantage

- This is a convenient way of storing a known number of elements of the same type.
- The memory allocation for data is done sequentially and does not engage any extra memory space.
- Accessing the arrays using their index value is a faster method.

Disadvantage

- The array allows storing only fixed number of element as per the size defined initially.
- If the actual number element is less than the declared size of an array the extra memory space allocated remains unutilized.
- Handling with the only same type of data also causes limitation at time.

~~04/09/25~~