Introduction to Arrays

CS201-Data Structure

Agenda

- Arrays
- Limitation of Arrays
- Dynamic Memory Allocation
- Dynamic Safe Array as ADT
- Jagged Array

Arrays

- A data structure, which stores a fixed-size sequential collection of elements of the same type. Eg. int a[]={10,20,30}
- Allocate one contiguous memory location
- Accessing array element

Array Declaration

//1-Dimension Arrays

- Syntax: data_type array_name[size];
- Int s[5]= {46, 41, 39, 38, 43}

//2-Dimension Arrays

- list of one dimension array
- Syntax: data_type array_name[rows][cols];
- Int s[3][5] = {{10,30,20,23,45}, {11,22,56,76,34}, {35,36,89,31,87}}

Static Array

```
int main()
{
 int m[5],i=0;
 for (int i=0;i<4;i++) {
   m[i] = i*2;
   cout<<i<<‘’element is’’<<m[i];
}
}
```

Static Memory Allocation :

The size of arrays must be determined during compile time and program can not change the size of the arrays.

Limitations of Arrays

Static Data

- Array is Static data Structure
- Memory Allocated during Compile time
- Once Memory is allocated at Compile Time it Cannot be Changed during Run-time

Inserting data in Array is Difficult

- Inserting element is very difficult because before inserting element in an array we have to create empty space by shifting other elements one position ahead

- This operation is faster if the array size is smaller, but same operation will be more and more time consuming and non-efficient in case of array with large size

Deletion Operation is difficult

- Deletion is not easy because the elements are stored in contiguous memory location.

- Like insertion operation , we have to delete element from the array and after deletion empty space will be created and thus we need to fill the space by moving elements up in the array.

Bound Checking

- Process of Checking the extreme limit of array is called Bound Checking and C++ does not perform Bound Checking.

- If the array range exceeds then we will get garbage value as result.

Shortage of Memory

- Array is Static data structure. Memory can be allocated at compile time only Thus if after executing program we need more space for storing additional information then we cannot allocate additional space at run time.
- Shortage of Memory , if we don’t know the size of memory in advance

Wastage of Memory

- Wastage of Memory , if array of large size is defined.

Dynamic Memory Allocation

- Using the keyword new to allocate.
- Usage of double pointers and single pointers.
- Memory required to store an array (in Bytes) = rows * columns * number_of_bytes_in_type.

Dynamically memory allocate and deallocate to array

- //1-Dimension dynamic Array allocation

```
int *data;
data = new int[size];
```

- //deallocation

```
delete [] data;
```

2d Dynamic Array

- //2-Dimension Dynamic Array allocation

```
Int **data;
data = new int*[nrows];
for( int i = 0 ; i < nrows ; i++ ){
        data[i] = new int [ncols];
    }
```

Rule of Three

- The Rule of Three is really two rules:

  (1) If a class has a nonempty destructor, it almost always needs a copy constructor and an assignment operator.

  (2) If a class has a nontrivial copy constructor or assignment operator, it usually needs both of these members and a destructor as well.

Vectors in c++

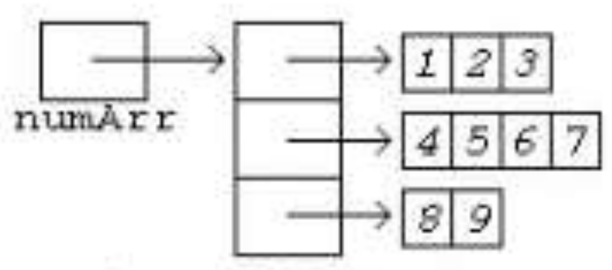
- Supports the functionality of same as Dynamic Arrays.
- Resize automatically when an element is inserted or deleted
- Include STL
- More powerful and generic than arrays
- Syntax vector <array-type> name-of-array

Safe Array

- out-of-bound check

Jagged Array

- Jagged array is an array of arrays in which the member arrays can be in different size.
- in contrast to the ordinary multi-dimensional array each row of the jagged array consist of different sizes. While in normal arrays it’s row size is constant as a we assign.



Layout of the same array