Dynamic Array

- Creat list [l]
- len [l]
- append [l]
- print [l]
- indexing [l]
- pop [l]
- find [l]
- insert [l]
- delete [l]
- remove [l]
- sort [l]
- min [l]
- max [l]

In [1]:
```
import ctypes #ctypes is a foreign function for python
```

In [2]:
```
class MeraList:
    def __init__(self):
        self.size=1
        self.n=0
        # creat a C type array with size = self.size
        self.A=self.__make_array(self.size)
    #Length
    def __len__(self):
        return self.n
    #Print
    def __str__(self):
        #[1,2,3]
        result=''
        for i in range(self.n):
            result=result+str(self.A[i])+','
        return '['+result[:-1]+']'
    #Indexing
    def __getitem__(self,index):
        if 0<=index<self.n:
            return self.A[index]
        else:
            return 'IndexError- Index out of range'
    #Append
    def append(self,item):
        if self.n==self.size:
            #resize
            self.__resize(self.size*2)

        self.A[self.n]=item
        self.n=self.n + 1
    #Pop
    def pop(self):
        if self.n==0:
            return 'Empty list'
```

```
        print(self.A[self.n-1])
        self.n=self.n-1
    #Clear
    def clear(self):
        self.n=0
        self.size=1

    #Find
    def find(self,item):
        for i in range(self.n):
            if self.A[i]==item:
                return i
        return 'ValueError not in list'
    #Remove
    def remove(self,item):
        pos = self.find(item)

        if type(pos)==int:
            #delete
            self.__delitem__(pos)
        else:
            return pos
    #Insert
    def insert(self,pos,item):
        if self.n==self.size:
            self.__resize(self.size*2)

        for i in range(self.n,pos,-1):
            self.A[i] = self.A[i-1]

        self.A[pos]=item
        self.n=self.n+1

    #Delete
    def __delitem__(self,pos):
        if 0<= pos<self.n:
            for i in range(pos,self.n-1):
                self.A[i]=self.A[i+1]

            self.n=self.n-1
    #Sort
    def sort(self):
        for i in range(self.n):
            for j in range(i, self.n):
                if self.A[j] < self.A[i]:
                    self.A[i], self.A[j] = self.A[j], self.A[i]
    #Min
    def min(self):
        if self.n == 0:
            raise ValueError('List is empty')
        min_val = self.A[0]
        for i in range(1, self.n):
            if self.A[i] < min_val:
                min_val = self.A[i]
        return min_val

    #Max
    def max(self):
        if self.n == 0:
```

```
            return None

        max_item = self.A[0]
        for i in range(1, self.n):
            if self.A[i] > max_item:
                max_item = self.A[i]

        return max_item


    #Resize
    def __resize(self,new_capacity):
        #creat a new array with new capacity
        B = self.__make_array(new_capacity)
        self.size=new_capacity
        #copy the content of A to B
        for i in range(self.n):
            B[i]=self.A[i]
        #reassign A
        self.A=B

    #Make Array
    def __make_array(self,capacity):
        #creats a C type array(static,refrential) with size capacity
        return (capacity*ctypes.py_object)()
```

In [3]:
```
l=MeraList()
```

In [4]:
```
l.append(5)
l.append(3.4)
l.append(6)
l.append(8)
l.append(63)
l.append(4)
l.append(9)
```

In [5]:
```
print(l)
```

```
[5,3.4,6,8,63,4,9]
```

In [6]:
```
l.max()
```

Out[6]:
```
63
```

In [7]:
```
l.min()
```

Out[7]:
```
3.4
```

In [8]:
```
l.sort()
print(l)
```

```
[3.4,4,5,6,8,9,63]
```

In [9]:
```
l.remove(63)
print(l)
```

```
[3.4,4,5,6,8,9]
```

In [10]:
```
del l[0]
print(l)
```

```
[4,5,6,8,9]
```

In [12]:
```
l.insert(0,7)
print(l)
```

```
[7,4,5,6,8,9]
```

In [13]:
```
l.find(9)
```

Out[13]:
```
5
```

In [14]:
```
l.pop()
```

```
9
```

In [15]:
```
l[2]
```

Out[15]:
```
5
```

In [16]:
```
l.append(89)
print(l)
```

```
[7,4,5,6,8,89]
```

In [17]:
```
print(len(l))
```

```
6
```

In [ ]:
```

```