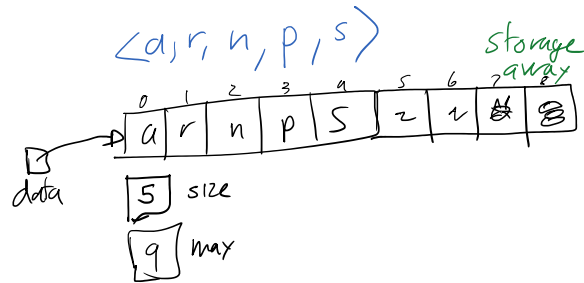


code;



implement a List in Code.

class Krustyburger, Donkey, Pant, Hower.

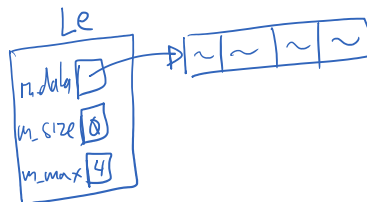
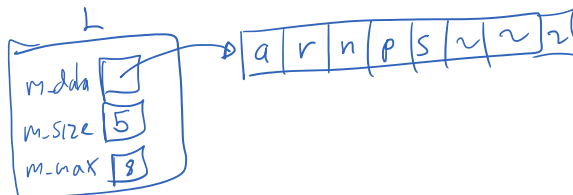


```

template <typename T>
class ArrayList {
    T* m_data;
    int m_size;
    int m_max;
}

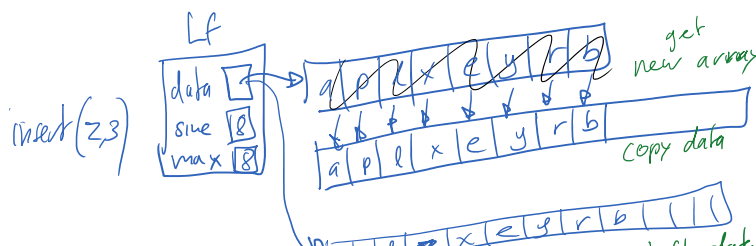
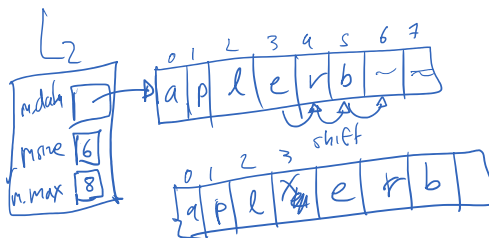
int ArrayList::size() const
{
    return m_size;
}

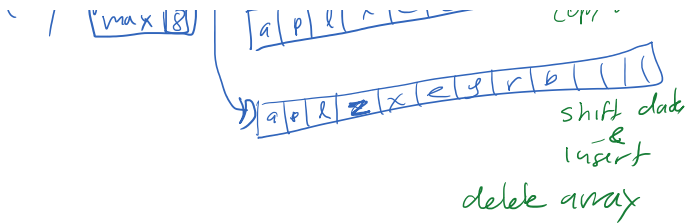
T& ArrayList::first() const
{
    if ( m_size > 0 )
        return m_data[0];
    else
        // OOPS!!!
}
    
```



• insert(x, 3)
 index must be < size.

- shift elements at pos ≥ i
- insert new data
- update m_size.





- insert(x, i)

if $i \leq m_size$ && $i > 0$

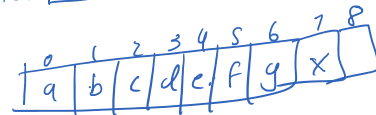
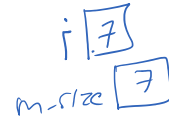
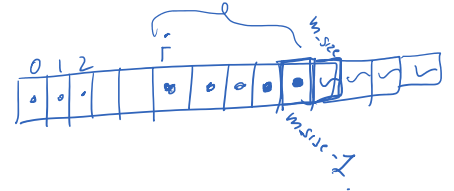
if array is full
 get a new array
 copy data to new array
 update m_max
 trash old array

shift elements at pos $\geq i$

insert x

update m_size

```
void ArrayList::insert(T& x, int i)
{
  if ( i >= 0 && i <= m_size ){
    if ( m_size == m_max ){
      T* tmp;
      tmp = new T[ m_max*2 ];
      for(int k=0; k<m_size; k++){
        tmp[k] = m_data[k];
      }
      delete [] m_data;
      m_data = tmp;
      m_max = m_max*2;
    }
    for( int k = m_size; k > i; k-- )
      m_data[k] = m_data[k-1];
    m_data[i] = x;
    m_size++;
  }
}
```



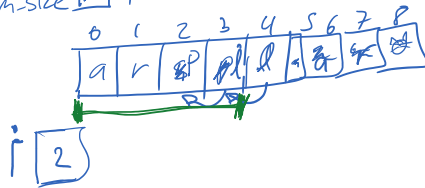
- C.A.R Hoare
- D. Knuth

"Premature optimisation is the root of all programming evil"

remove(i)



$i \in [0..m_size-1]$



$m_size > 0$



remove(i)

if list is not empty and i is in bounds then

- shift every element at pos $> i$ (to the left)

- decrement m_size.

- if storage array too big
 - get new smaller array

```
void ArrayList::remove( int i ) {
  if ( m_size > 0 && i >= 0 && i < m_size ){
    for (int k = i; k < m_size-1; k++){
      m_data[k] = m_data[k+1];
    }
    m_size--;

    if( m_max > m_size*4 ){
      T* tmp = new T[m_max/2];
      for(int k=0; k < m_size; k++){
        tmp[k] = m_data[k];
      }
      delete [] m_data;
      m_data = tmp;
      m_max = m_max/2;
    }
  }
}
```



- if storage array too big
- get new smaller array
- copy data to new array
- delete old array
- update m_max

```
T* tmp = new T[m_max/2];
for(int k=0; k < m_size; k++)
    tmp[k] = m_data[k];
delete [] m_data;
m_data = tmp;
m_max = m_max/2;
}
}
```

