

Priority Queues

Fragment of Textbook Code

```
public class BinaryHeap<AnyType extends
    Comparable<? super AnyType>>
{
    private static final int DEFAULT_CAPACITY = 10;

    private int currentSize;    // Number of elements in heap
    private AnyType [ ] array; // The heap array

    public BinaryHeap( ){}
    public BinaryHeap( int capacity ){}
    public void insert( AnyType x ){}
    public AnyType deleteMin( ){}
}
```

```
public BinaryHeap( )  
{  
    this( DEFAULT_CAPACITY );  
}
```

```
public BinaryHeap( int capacity )  
{  
    currentSize = 0;  
    array = (AnyType[]) new Comparable[ capacity + 1 ];  
}
```

```
public void insert( AnyType x )
{
    if( currentSize == array.length - 1 )
        enlargeArray( array.length * 2 + 1 );

    // Percolate up
    int hole = ++currentSize;
    for( ; hole > 1 && x.compareTo( array[ hole / 2 ] ) < 0;
        hole /= 2 )
        array[ hole ] = array[ hole / 2 ];
    array[ hole ] = x;
}
```

```
private void enlargeArray( int newSize )
{
    AnyType [] old = array;
    array = (AnyType []) new Comparable[ newSize ];
    for( int i = 0; i < old.length; i++ )
        array[ i ] = old[ i ];
}
```

```
public AnyType deleteMin( )
{
    if( isEmpty( ) )
        throw new UnderflowException( );

    AnyType minItem = array[1];
    array[ 1 ] = array[ currentSize-- ];
    percolateDown( 1 );

    return minItem;
}
```

```
private void percolateDown( int hole )  
{  
    int child;  
    AnyType tmp = array[ hole ];
```

```
for( ; hole * 2 <= currentSize; hole = child )
{
    child = hole * 2;
    if( child != currentSize &&
        array[ child + 1 ].compareTo( array[ child ] ) < 0 )
        child++;
    if( array[ child ].compareTo( tmp ) < 0 )
        array[ hole ] = array[ child ];
    else
        break;
}
array[ hole ] = tmp;
}
```



```
public static void main( String [ ] args )
{
    int numItems = 10000;
    BinaryHeap<Integer> h = new BinaryHeap<Integer>( );
    int i = 37;

    for( i = 37; i != 0; i = ( i + 37 ) % numItems )
        h.insert( i );
    for( i = 1; i < numItems; i++ )
        if( h.deleteMin( ) != i )
            System.out.println( "Oops! " + i );
}
```