

## MinHeap.java

```

1 /*
2  * Written by JJ Shepherd
3  */
4 public class MinHeap <T extends Comparable<T>>{
5     private T[] heap;
6     public static final int DEF_SIZE = 128;
7     private int lastIndex;//First null element
8     public MinHeap()
9     {
10         init(DEF_SIZE);
11     }
12     public MinHeap(int size)
13     {
14         init(size);
15     }
16     private void init(int size)
17     {
18         if(size > 0)
19         {
20             heap = (T[])(new Comparable[size]);
21             lastIndex = 0;
22         }
23         else
24             init(DEF_SIZE);
25     }
26     public void add(T aData)
27     {
28         if(lastIndex >= heap.length)
29             return;
30         heap[lastIndex] = aData;
31         bubbleUp();
32         lastIndex++;
33     }
34     private void bubbleUp()
35     {
36         int index = lastIndex;
37         while(index > 0 && heap[(index-1)/2].compareTo(heap[index])>0)
38         {
39             T temp = heap[(index-1)/2];
40             heap[(index-1)/2] = heap[index];
41             heap[index] = temp;
42             index = (index-1)/2;
43         }
44     }
45     public T remove()
46     {
47         if(lastIndex == 0)
48             return null;
49         T ret = heap[0];
50         heap[0] = heap[lastIndex-1];
51         heap[lastIndex-1] = null;
52         lastIndex--;
53         bubbleDown();
54         return ret;
55     }
56     private void bubbleDown()
57     {

```

## MinHeap.java

```
58     int index = 0;
59     while(index*2+1 < lastIndex)
60     {
61         int smallIndex = index*2+1;
62         if(index*2+2 < lastIndex && heap[index*2+1].compareTo(heap[index*2+2])>0)
63             smallIndex = index*2+2;
64         if(heap[index].compareTo(heap[smallIndex])>0)
65         {
66             T temp = heap[index];
67             heap[index] = heap[smallIndex];
68             heap[smallIndex] = temp;
69         }
70         else
71             break;
72         index = smallIndex;
73     }
74 }
75 public void print()
76 {
77     for(T h : heap)
78     {
79         if(h == null)
80             break;
81         System.out.println(h);
82     }
83 }
84 public void heapSort(T[] array)
85 {
86     if(array == null)
87         return;
88     MinHeap<T> mHeap = new MinHeap<T>(array.length);
89     for(int i=0;i<array.length;i++)
90         mHeap.add(array[i]);
91     for(int i=0;i<array.length;i++)
92         array[i] = mHeap.remove();
93 }
94 }
95 }
96
```