```
import java.util.ArrayList;
/**
 *
 * @author Rasmus Bartholin og Mads Mikael Keinicke
 * Rasmus: rbart17
 * Mads: makei17
 */
public class PQHeap implements PQ {
    //Field
    private final int MaxElms;
    private final ArrayList<Element> PrioArray;
    private int ListSize;

    // Constructor
    public PQHeap(int MaxElms)
    {
        this.MaxElms = MaxElms;
        this.ListSize = -1;
        this.PrioArray = new ArrayList<>(MaxElms);
    }

    private int Parent(int i){
        return i/2;
    }


    private Element heapMinimum(){
        return this.PrioArray.get(0);
    }


    @Override
    public void insert(Element key){

        ListSize++;
        int i = ListSize;
        this.PrioArray.add(key);
        while( i > 0 && this.PrioArray.get(Parent(i)).freq  >  this.PrioArray.get(i).freq){
            Element tmp = this.PrioArray.get(i);
            this.PrioArray.set(i, this.PrioArray.get(Parent(i)));
            this.PrioArray.set(Parent(i), tmp);
            // Den skal blive ved med at tjekke, fordi den nye forÃ¦ldre kan ogsÃ¥ vÃ¦re "svagere".
            i = Parent(i);

        }

    }


    public void printHeap()
    {
        for(Element x : PrioArray)
        {
            System.out.print(x.getFreq() + " ");
        }
        System.out.println();
    }
```

```
  private int left(int i){
    return 2*i + 1;
  }

  private int right(int i){
    return 2*i + 2;
  }



  @Override
  public Element extractMin(){
      if(this.ListSize < 0){
          return null;
      }
      else{

          Element min = this.PrioArray.get(0);
          //System.out.println("Extracing this:");
          //System.out.println(min.getFreq());
          this.PrioArray.set(0, this.PrioArray.get(this.ListSize));
          this.ListSize--;
          PrioArray.remove(PrioArray.size()-1);
          minHeapify(0);
          //System.out.println("List size = " + ListSize);
          //System.out.println("New smallest number");
          //Element tmp = PrioArray.get(0);
          //System.out.println(tmp.getFreq());
          return min;

      }

  }

  private void exchange(int a, int b){
      Element aa = this.PrioArray.get(a);
      Element bb = this.PrioArray.get(b);
      this.PrioArray.set(a, bb);
      this.PrioArray.set(b, aa);
  }

  private void minHeapify(int i){
      int l = left(i);
      int r = right(i);

      int smallest;
      if (l <= this.ListSize && this.PrioArray.get(l).freq < this.PrioArray.get(i).freq){
          smallest = l;
      } else {
          smallest = i;
      }
      if (r <= this.ListSize && this.PrioArray.get(r).freq < this.PrioArray.get(smallest).freq){
          smallest = r;
      }
      if (smallest != i){
          exchange(i, smallest);
          minHeapify(smallest);
      }
```

```
    }

    public int getSize()
    {
        int tmp = this.ListSize+1;
        return tmp;
    }


}
```