Queue<Integer> queue = new LinkedList<Integer>();

peek()

poll()

add(e)

```
static <E> List<E> heapSort(Collection<E> c) {
Queue<E> queue = new PriorityQueue<E>(c);
List<E> result = new ArrayList<E>();

while (!queue.isEmpty())
result.add(queue.remove());

return result;
}
```

ArrayList

boolean add(E e)

void add(int index, E element)

E remove(int index)

boolean remove(Object o)

void clear()

boolean addAll(Collection<? extends E> c)

E set(int index, E element)

```
ArrayList<Integer> list = new ArrayList<Integer>();

ArrayList()
```
Constructs an empty list with an initial capacity of ten.
```
ArrayList(Collection<? extends E> c)
```
Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.
```
ArrayList(int initialCapacity)
```
Constructs an empty list with the specified initial capacity.