```java
public class HeapSort {
    public static void sort(int[] array) {
        // Построение кучи (перегруппируем массив)
        for (int i = array.length / 2 - 1; i >= 0; i--)
            heapify(array, array.length, i);

        // Один за другим извлекаем элементы из кучи
        for (int i = array.length - 1; i >= 0; i--) {
            // Перемещаем текущий корень в конец
            int temp = array[0];
            array[0] = array[i];
            array[i] = temp;

            // Вызываем процедуру heapify на уменьшенной куче
            heapify(array, i, 0);
        }
    }

    private static void heapify(int[] array, int heapSize, int rootIndex) {
        int largest = rootIndex; // нициализируем наибольший элемент как
корень
        int leftChild = 2 * rootIndex + 1; // левый = 2*rootIndex + 1
        int rightChild = 2 * rootIndex + 2; // правый = 2*rootIndex + 2

        // Если левый дочерний элемент больше корня
        if (leftChild < heapSize && array[leftChild] > array[largest])
            largest = leftChild;

        // Если правый дочерний элемент больше, чем самый большой элемент на
данный момент
        if (rightChild < heapSize && array[rightChild] > array[largest])
            largest = rightChild;
        // Если самый большой элемент не корень
        if (largest != rootIndex) {
            int temp = array[rootIndex];
            array[rootIndex] = array[largest];
            array[largest] = temp;

            // Рекурсивно преобразуем в двоичную кучу затронутое поддерево
            heapify(array, heapSize, largest);
        }
    }
}
```