堆排序（Heap Sort）

- 堆排序可以认为是对选择排序的一种优化

- 执行流程

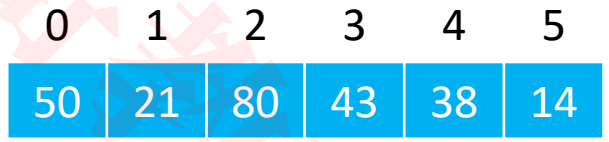
① 对序列进行原地建堆（heapify）

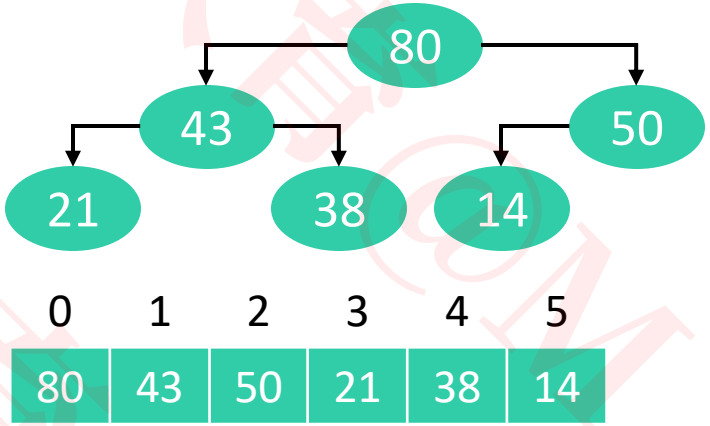
② 重复执行以下操作，直到堆的元素数量为 1

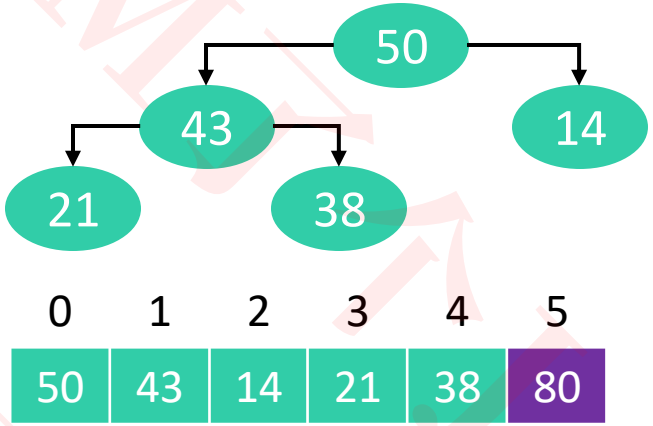
✓ 交换堆顶元素与尾元素

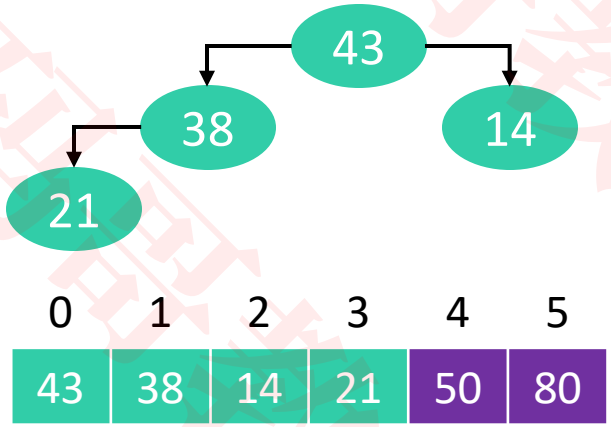
✓ 堆的元素数量减 1

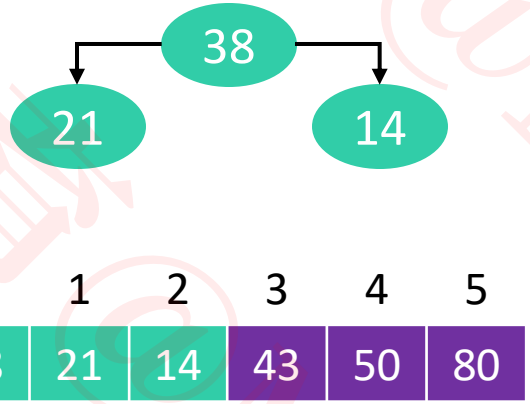
✓ 对 0 位置进行 1 次 siftDown 操作

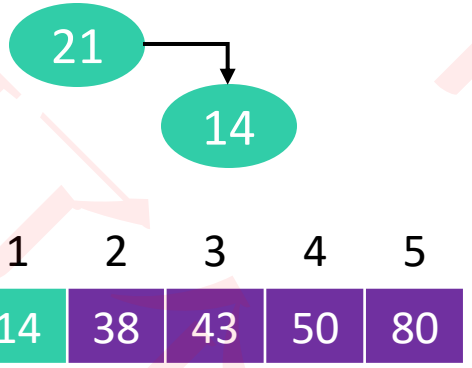


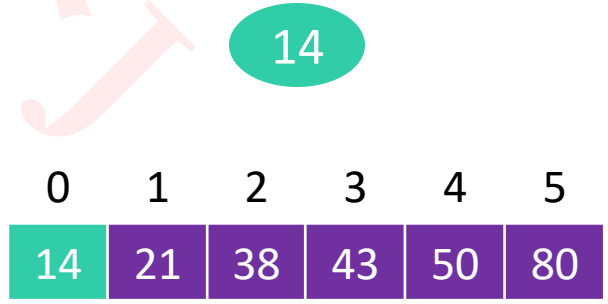












堆排序 – 实现

```
// heapify
heapSize = array.length;
for (int i = (heapSize >> 1) - 1; i >= 0; i--) {
    siftDown(i);
}

while (heapSize > 1) {
    swap(0, --heapSize);
    siftDown(0);
}
```

```
private void siftDown(int index) {
    T element = array[index];
    int half = heapSize >> 1;
    while (index < half) {
        int childIndex = (index << 1) + 1;
        T child = array[childIndex];
        int rightIndex = childIndex + 1;
        if (rightIndex < heapSize
                && cmp(array[rightIndex], child) > 0) {
            child = array[childIndex = rightIndex];
        }
        if (cmp(element, child) >= 0) break;
        array[index] = child;
        index = childIndex;
    }
    array[index] = element;
}
```

- 最好、最坏、平均时间复杂度：O(nlogn)，空间复杂度：O(1)，属于不稳定排序