

# 第三次作业

## 1 分析 SwapDown 和 heapify 两个函数的work和span

- work: heapify 是  $O(n)$ , 因为需要自底向上依次构建子树为最小堆, 需要遍历到每一个节点, swapDown 是  $O(\log n)$ , 因为每次 swap down 只用处理层数这么多次( $O(\log n)$ 次)即可
- span: heapify 和 swapDown 都是  $O(\log n)$ , 在并行处理的前提下复杂度都只和二叉树层数有关

## 2 分析 Msort 函数的 work 和 span

- work: 第  $m$  轮需要处理  $2^m$  组, 每组工作量是  $n * 2^{-m}$ , 一共需要  $\log n$  轮, 总 work 为  $n * \log n$
- span: 在并行处理的前提下同一轮的多个组可同时处理, 复杂度都只和层数有关, span 为  $O(\log n)$