

## INF 112 – Prática 6, *QuickSort* e *HeapSort* – 2019/2

**Exercício 0:** O *heapSort* que vimos em sala foi projetado para ordenar vetores que têm índices começando em 1. Em C/C++, vetores usualmente começam no índice 0. Faça uma modificação simples na função *heapSort* de forma que seja possível ordenar vetores que têm o primeiro elemento no índice 0. Sugestão: copie os elementos do vetor a ser ordenado para um novo vetor.

**Exercício 1:** Repita o Exercício 0, mas agora, ao invés de copiar os elementos para um novo vetor, proponha uma modificação das propriedades de um *heap* para os casos em que vetores tenham o primeiro elemento indexado em 0.

**Exercício 2:** Proponha um algoritmo, baseado no *heapSort*, que possibilite encontrar os  $k$  menores elementos de um vetor em tempo  $O(n + k \log(n))$ .

**Exercício 3:** Na aula sobre *quickSort*, foi mencionado que é interessante combinar o *quickSort* com o *InsertionSort*. A ideia é ignorar as partições pequenas geradas pelo *quickSort* e, após o final do algoritmo, utilizar o *insertionSort* no vetor quase ordenado. Implemente essa abordagem de ordenação. Após isso, gere um vetor aleatório de tamanho muito grande e compare o tempo de execução das duas abordagens.

**Exercício 4:** A escolha do pivô no *quickSort* é um passo importante. Como foi visto em sala, escolhas sistemáticas e ruins do pivô podem levar a um tempo quadrático. Nesse exercício, vamos investigar uma abordagem comum para evitar o pior caso do *quickSort*. Ao invés de definir o índice do pivô como sendo o primeiro elemento do subvetor a ser particionado, escolha três índices de forma aleatória (que estejam dentro do subvetor) e retorne o índice do elemento mediano dentre os três selecionados. Implemente essa abordagem do *quickSort* e a compare com a versão tradicional. Para a comparação, gere três entradas grandes, sendo a primeira já ordenada, a segunda em ordem inversa e a terceira aleatória. Após isso, combine seu novo *quickSort* com o *insertionSort* (como no Exercício 3). Novamente, compare o tempo de execução com o *quickSort* tradicional, que foi visto em sala.

**Exercício 5:** Chame o professor para mostrar o andamento de seu trabalho prático.