MC458 — PA1

Professor: Pedro J. de Rezende

Leonardo Rodrigues Marques 178610

Código - Completo
Relatório - Bom!
Nota total - 1.0

Busca Linear

A Busca Linear em si tem complexidade $\theta(n)$. Em casos pequenos, $k \in 0, 1, 2...$, o algoritmo realmente performa de forma linear. A medida que o k aumenta, da ordem de **n**, observamos que o algoritmo perde performance. Ou melhor dizendo, $k \in O(n)$, portanto temos uma busca linear em termos de n^2, um grau de baixa eficiência.

Quick Sort

O Quick Sort é um algoritmo com complexidade de pior caso $O(n^2)$. Apesar da complexidade de pior caso ser desfavorável, ele possui uma complexidade de caso médio $O(n \log\ n)$, o que proporciona desempenho consistente em toda faixa de valores de k. Existe a possibilidade de em um determinado valor de k dado um vetor V caia em solução $O(n^2)$, entretanto, não é regra.

Min-Heap

O Min-Heap possui complexidade em torno de O(n log n). Ele se comporta dessa em valores de k pequenos. Entretanto, a medida que k que tende ao tamanho do vetor n, vemos um incremento na complexidade **n** na complexidade, portanto resultando em termos de O(n + n log n). Apesar de se comportar de forma bem consistente (de qualquer forma se mostra como O(n log n)), em valores de de **n** grandes, ele pede um pouco de desempenho.

Análise Prática

Os métodos 1, 2 e 3 possuem tempos de execução que delimitam sua melhor performance em determinados ks. O método 1, que é a busca linear, apresentou um desempenho satisfatório em ks pequenos, pois como analisado, sua execução nessa situação é de ordem n. A partir de um certo valor, observamos que o min-Heap entra em cena. Com pior caso n log n e valores de k medianos, ele consegue executar com maior eficiência comparados aos outros dois algoritmos. Mas a medida que o k aumenta drasticamente, um valor grande é adicionado a sua complexidade. Apesar do valor ser linear, o impacto é refletido no tempo de execução dado a sua magnitude. E por fim, temos o Quick Sort. Esse algoritmo performa de forma

consistente. Enquanto a busca linear e o min-Heap perde eficiência a medida que o k aumenta, o Quick Sort mantém relativamente constante no seu tempo de execução. Dessa forma, para k maiores, é desejável usar Quick Sort (exceto pro pior caso, que não foi observado na prática).

Valores

- $K_1 = 8$
- $K_2 = 513810$