Relatório de Análise de Algoritmos

Daniel Marques, Miguel Brito, Jefferson Oliveira, Vinicius Gonzaga June 29, 2017

1 Introdução

1.1 Counting Sort

Analise

O quicksort é um algoritmo de ordenação por divisão e conquista. Seu funcionamento está relacionado diretamente ao particionamento do vetor, que consiste em selecionar um elemento pivô e separar todos os elementos que sejam menores que esse pivo de todos os elementos que sejam maiores, em duas partições separadas. Recursivamente realiza o particionamento, até que se atinja o caso base.

- Tempo no melhor caso: $\theta(nlgn)$
- Tempo no pior caso: $O(n^2)$
- Tempo no caso médio: $\theta(nlgn)$

2 Tempos

Figure 1: Heapsort vetor aleatório

Método QuickSort com vetor aleatorio T(n) = n*15.59Log(n) + 1692

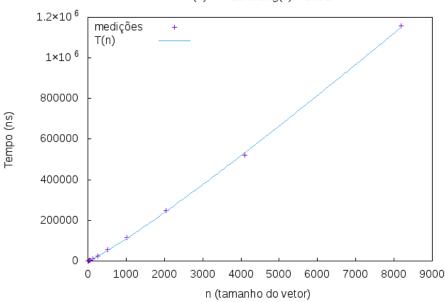


Figure 2: Heapsort vetor crescente

Método QuickSort com vetor em ordem crescente T(n) = 1.47n**2 + 55.16n + -5411

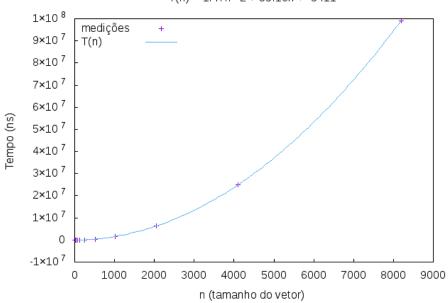


Figure 3: Heapsort vetor decrescente

Método QuickSort com vetor em ordem decrescente T(n) = 1.496n**2 + 237n + -3.484e+04

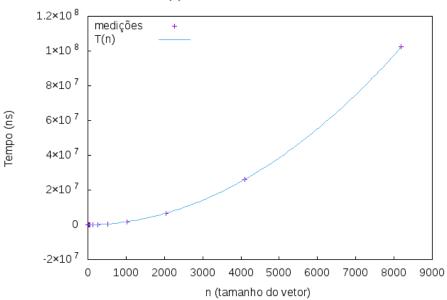


Figure 4: Heapsort vetor parcialmente crescente 60 %

Método QuickSort com vetor parcialmente em ordem crescente(60) T(n) = 0.002853n**2 + 139.6n + -6346

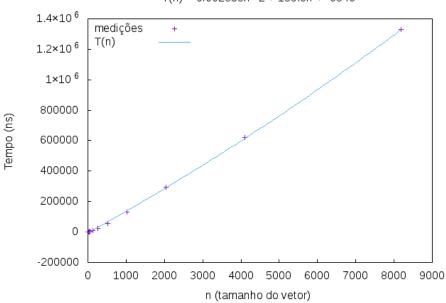


Figure 5: Heapsort vetor parcialmente crescente 70 %

Método QuickSort com vetor parcialmente em ordem crescente(70) T(n) = 0.0001523n**2 + 170.4n + -1.181e+04

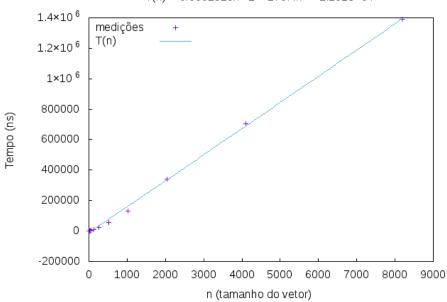


Figure 6: Heapsort vetor parcialmente crescente 80 %

Método QuickSort com vetor parcialmente em ordem crescente(80) T(n) = -0.002539n**2 + 229.7n + -1.844e+04

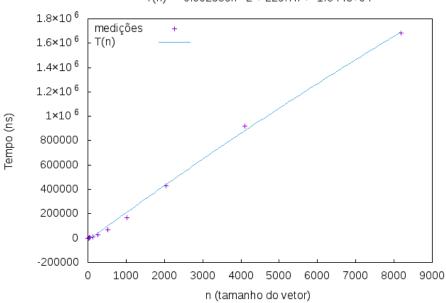


Figure 7: Heapsort vetor parcialmente crescente 90 %

Método QuickSort com vetor parcialmente em ordem crescente(90) T(n) = 0.008174n**2 + 268.9n + -2.346e+04

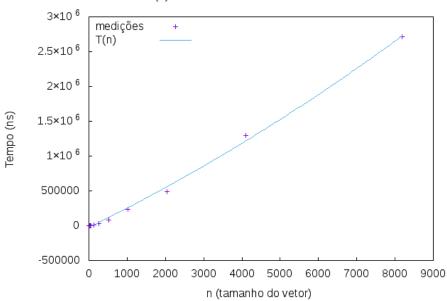


Figure 8: Heapsort vetor parcialmente decrescente 60 %

Método QuickSort com vetor parcialmente em ordem decrescente(60) T(n) = 0.004195n**2 + 101.8n + -2567

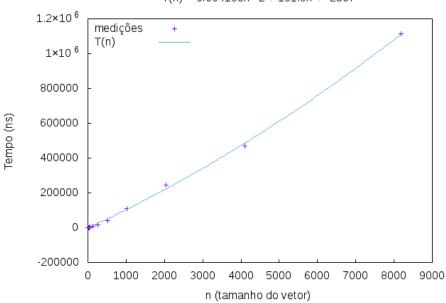


Figure 9: Heapsort vetor parcialmente decrescente 70 %

Método QuickSort com vetor parcialmente em ordem decrescente(70) T(n) = 0.002453n**2 + 128.1n + -1.089e+04

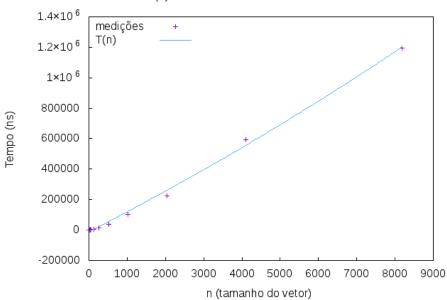


Figure 10: Heapsort vetor parcialmente decrescente 80 %

Método QuickSort com vetor parcialmente em ordem decrescente(80) T(n) = 0.008449n**2 + 101.8n + -3582

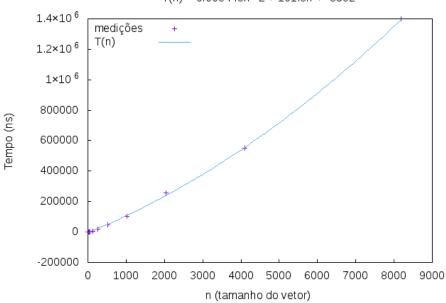


Figure 11: Heapsort vetor parcialmente decrescente 90 %

Método QuickSort com vetor parcialmente em ordem decrescente(90) T(n) = 0.002062n**2 + 173.2n + -1.336e+04

