

# Relatório de Análise de Algoritmos

Daniel Marques, Miguel Brito, Jefferson Oliveira, Vinicius Gonzaga

June 29, 2017

## 1 Introdução

### 1.1 Quick Sort

Analise

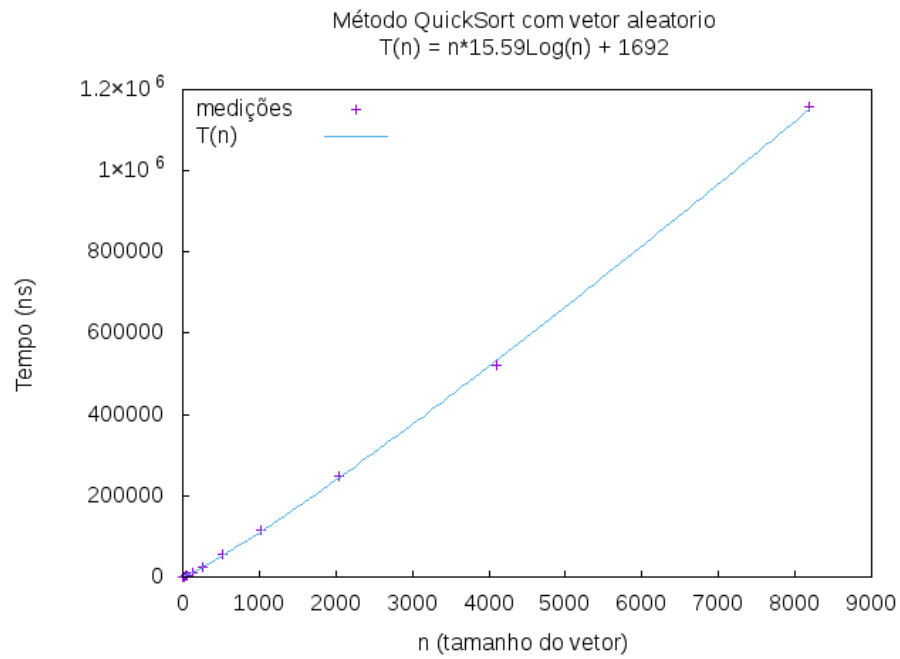
O Quicksort é um algoritmo de ordenação por divisão e conquista. Seu funcionamento está relacionado diretamente ao particionamento do vetor, que consiste em selecionar um elemento pivô e separar todos os elementos que sejam menores que esse pivô de todos os elementos que sejam maiores, em duas partições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

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

## 2 Tempos

### 2.1 Vetores aleatórios

Figure 1: Quicksort vetor aleatório



Para  $n = 2^{32}$ :

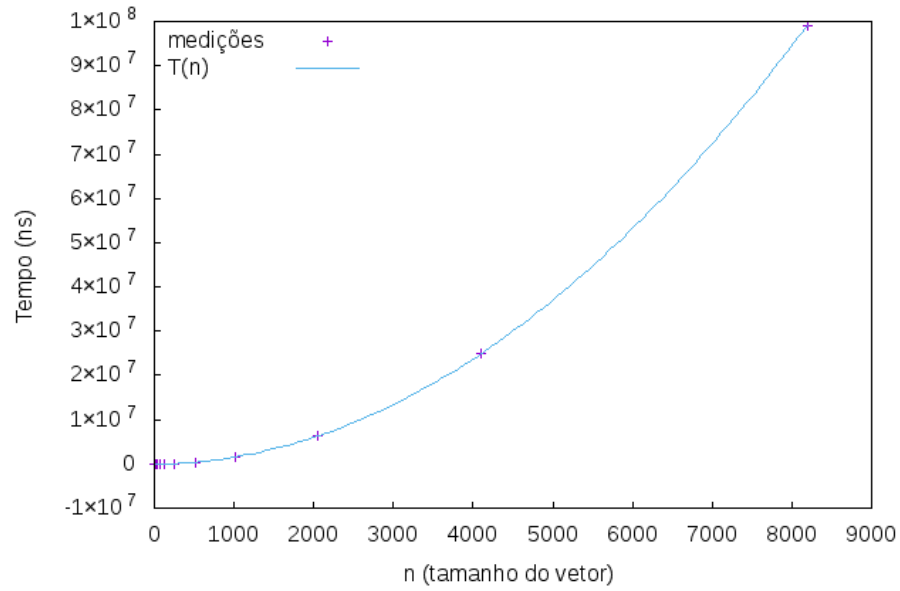
$$T(n) = 2^{32} * 15.59 * \lg 2^{32} + 1692 = 2.1426733e+12 \text{ ns} = 2142 \text{ s} = 35,7 \text{ min.}$$

## 2.2 Vetores totalmente crescentes

Figure 2: Quicksort vetor crescente

Método QuickSort com vetor em ordem crescente

$$T(n) = 1.47n^2 + 55.16n - 5411$$

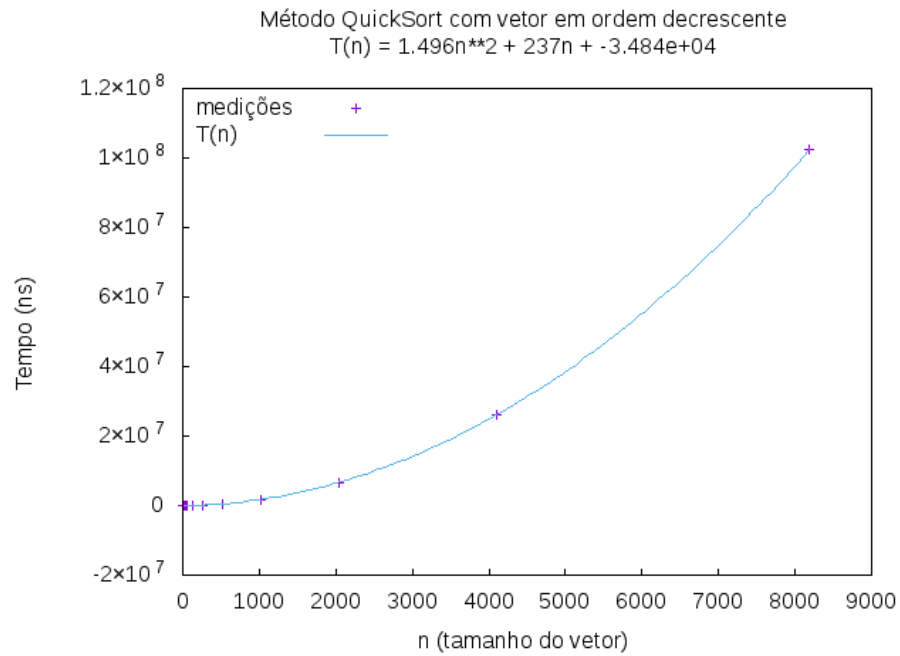


Para  $n = 2^{32}$ :

$$T(n) = 2^{64} * 1.47 + 55.16 * 2^{32} - 5411 = 2.7116714e+19 \text{ ns} = 2.7116714e+10 \text{ s} = 8,59 \text{ sécs.}$$

## 2.3 Vetores totalmente decrescentes

Figure 3: Quicksort vetor decrescente



Para  $n = 2^{32}$ :

$$T(n) = 2^{64} * 1.496 + 237 * 2^{32} - 3480 = 2.7596329e+19 \text{ ns} = 2.7596329e+10 = 8,75 \text{ s\u00e9cs.}$$

## 2.4 Vetores parcialmente crescentes

Figure 4: Quicksort vetor parcialmente crescente 60 %

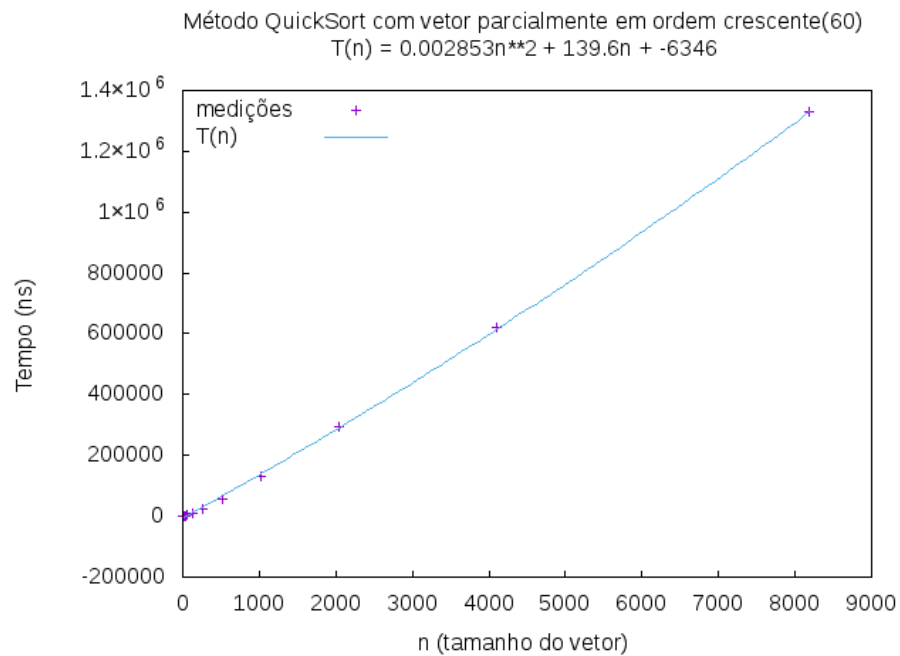


Figure 5: Quicksort vetor parcialmente crescente 70 %

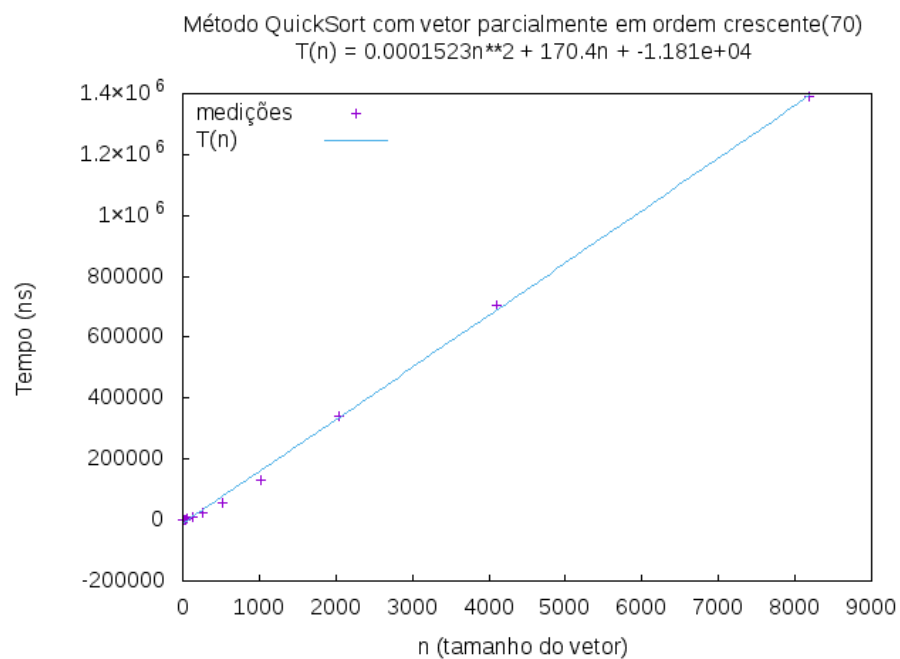


Figure 6: Quicksort vetor parcialmente crescente 80 %

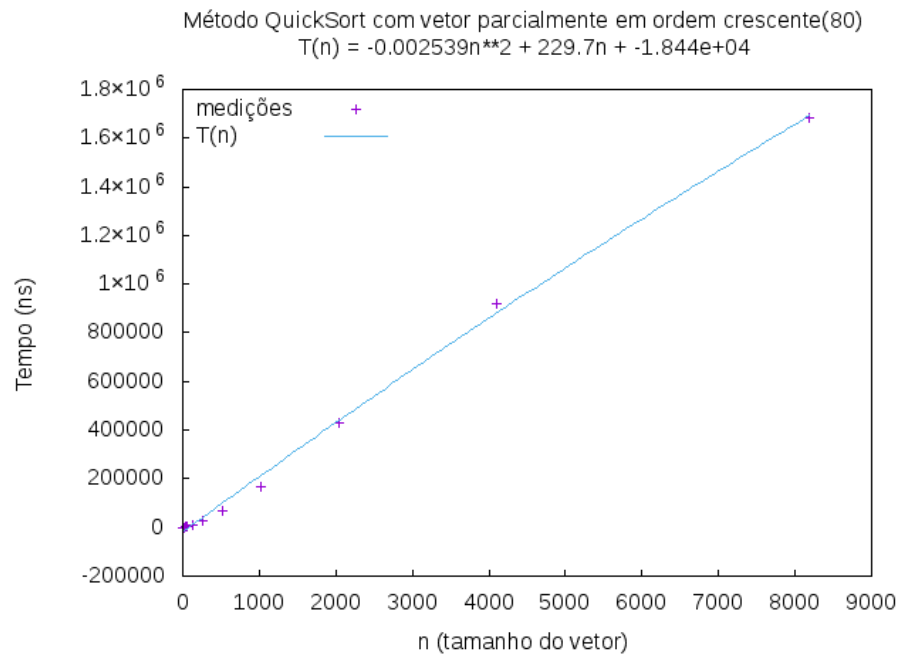
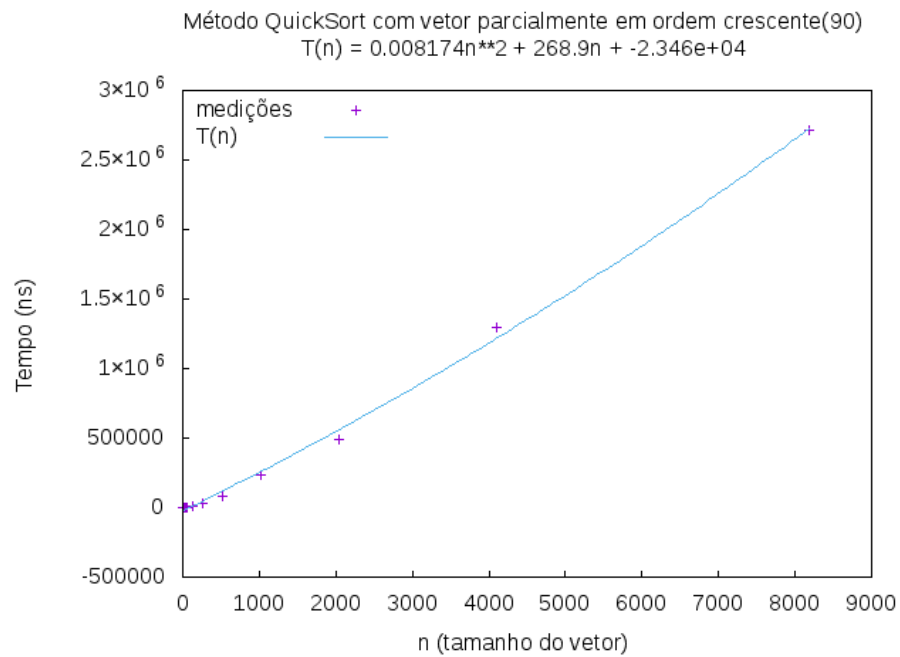


Figure 7: Quicksort vetor parcialmente crescente 90 %



## 2.5 Vetores parcialmente decrescentes

Figure 8: Quicksort vetor parcialmente decrescente 60 %

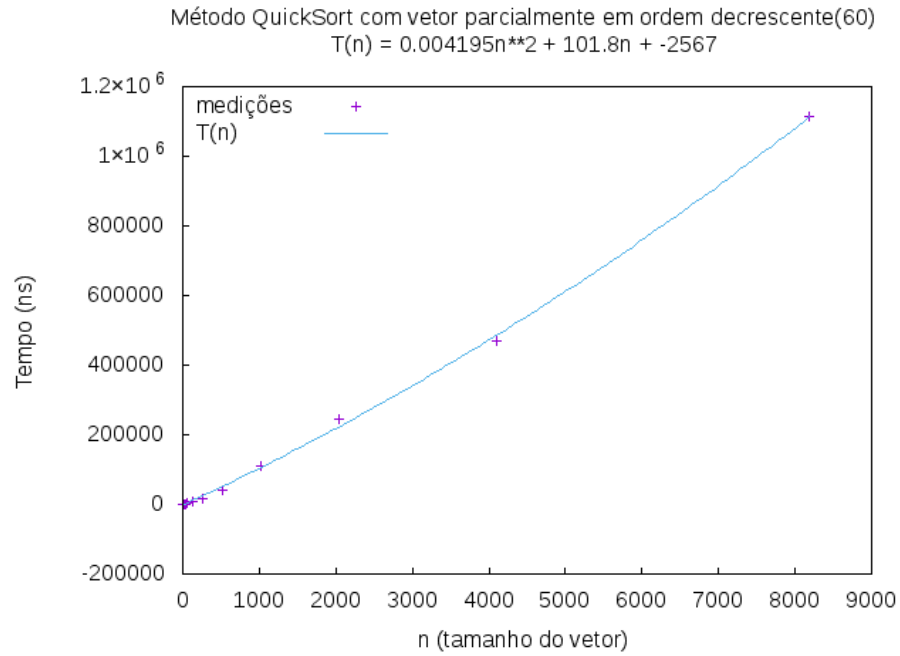


Figure 9: Quicksort vetor parcialmente decrescente 70 %

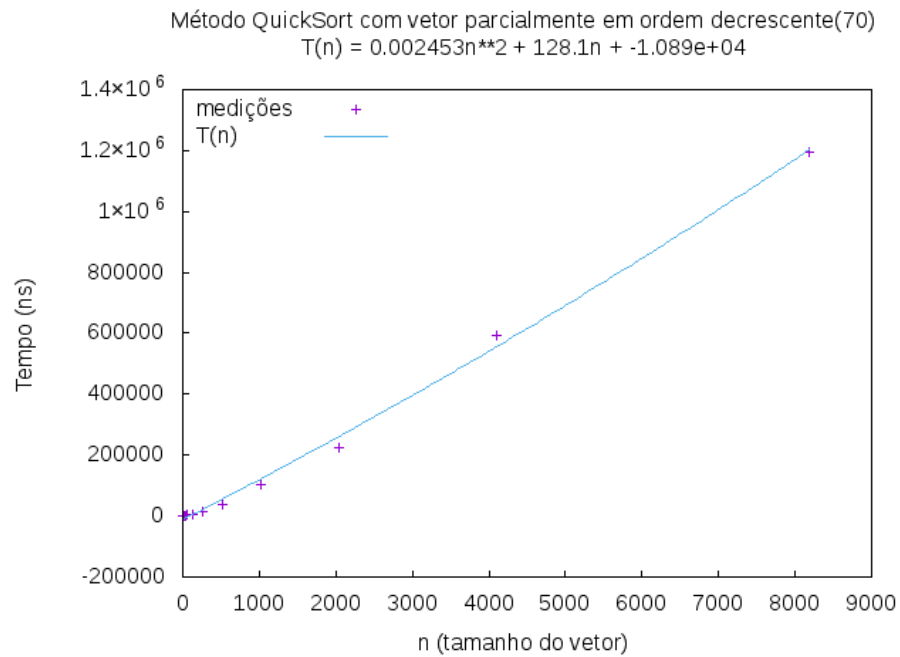


Figure 10: Quicksort vetor parcialmente decrescente 80 %

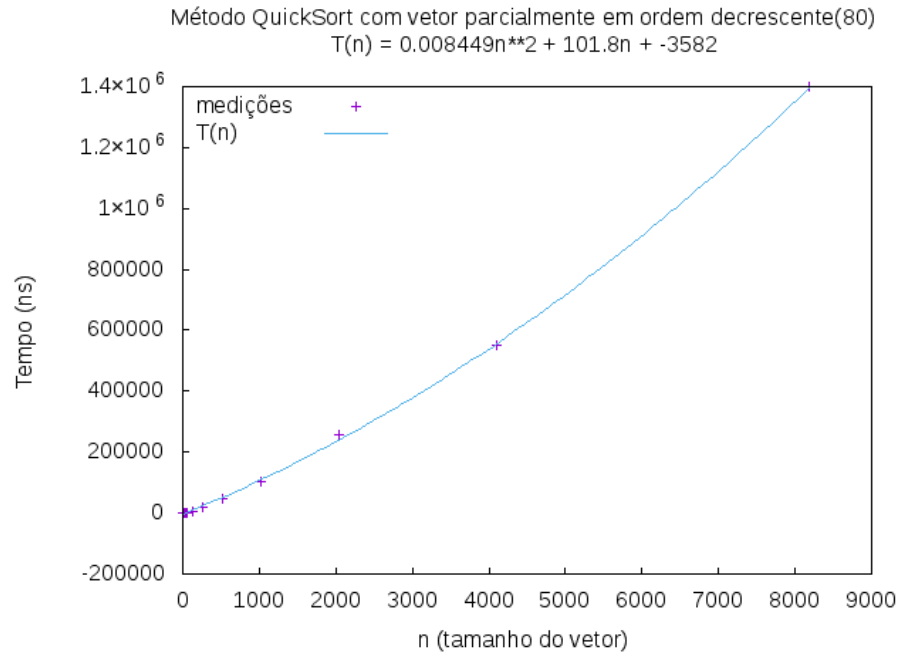


Figure 11: Heapsort vetor parcialmente decrescente 90 %

