## Relatório de Análise de Algoritmos

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

## 1 Introdução

## 1.1 Counting Sort

Analise

O heapsort é um algoritmo de ordenação que se utilizada da estrutura de dados heap. Seu funcionamento consiste em criar um heap de máximo a partir do vetor original, em seguida trocar o primeiro elemento com o último e decrementar o tamanho do heap. Em seguida novamente cria-se um heap de máximo com a nova raiz. Dessa forma segue-se ordenando o vetor, do último para o primeiro elemento.

- Tempo no melhor caso:  $\theta(nlgn)$
- Tempo no pior caso:  $\theta(nlgn)$
- Tempo no caso médio:  $\theta(nlgn)$

## 2 Tempos

Figure 1: Heapsort vetor aleatório

Método HeapSort com vetor aleatorio T(n) = n\*39.3Log(n) + 2.294e+04

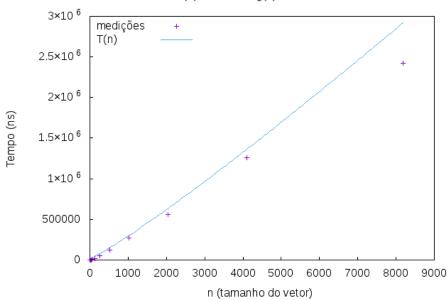


Figure 2: Heapsort vetor crescente

Método HeapSort com vetor crescente T(n) = n\*35.05Log(n) + 0.46

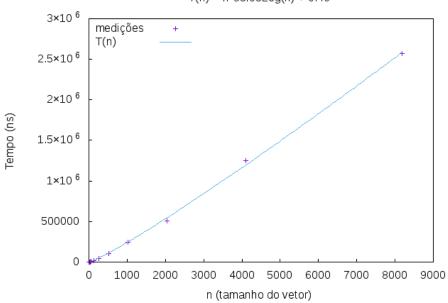


Figure 3: Heapsort vetor decrescente

Método HeapSort com vetor decrescente T(n) = n\*35.24Log(n) + -9700

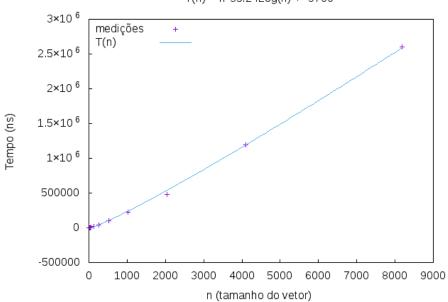


Figure 4: Heapsort vetor parcialmente crescente 60 %

Método HeapSort com vetor parcialmente crescente(60) T(n) = n\*36.76Log(n) + 6325

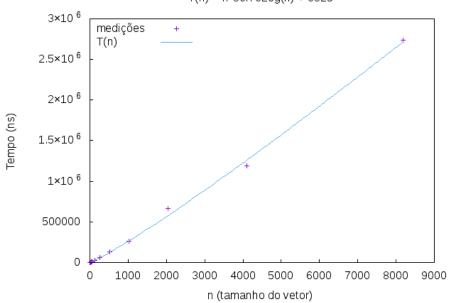


Figure 5: Heapsort vetor parcialmente crescente 70 %

Método HeapSort com vetor parcialmente crescente(70) T(n) = n\*37.58Log(n) + 8084

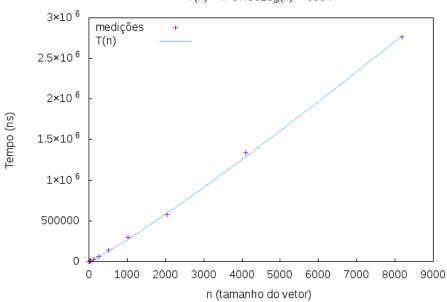


Figure 6: Heapsort vetor parcialmente crescente 80 %

Método HeapSort com vetor parcialmente crescente(80) T(n) = n\*34.43Log(n) + -2923

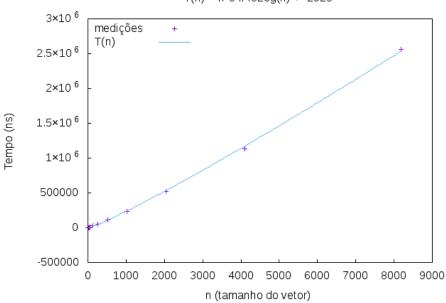


Figure 7: Heapsort vetor parcialmente crescente 90 %

Método HeapSort com vetor parcialmente crescente(90) T(n) = n\*35.17Log(n) + 6857

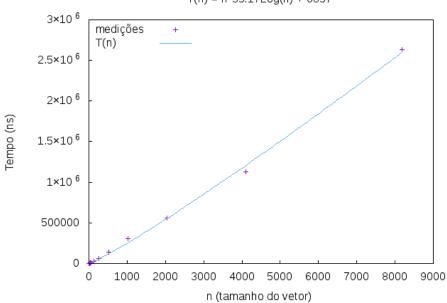


Figure 8: Heapsort vetor parcialmente decrescente 60 %

Método HeapSort com vetor parcialmente decrescente(60) T(n) = n\*36.76Log(n) + 1.553

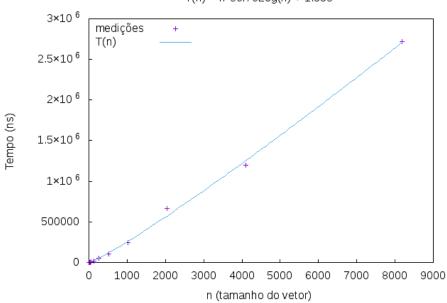


Figure 9: Heapsort vetor parcialmente decrescente 70 %

Método HeapSort com vetor parcialmente decrescente(70) T(n) = n\*35.75Log(n) + 2.151

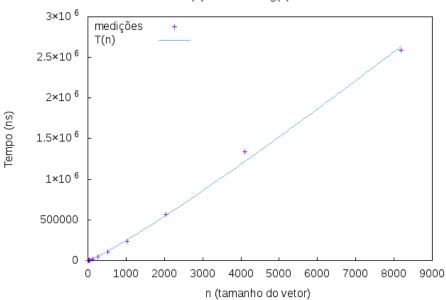


Figure 10: Heapsort vetor parcialmente decrescente 80 %

Método HeapSort com vetor parcialmente decrescente(80) T(n) = n\*34.76Log(n) + -4287

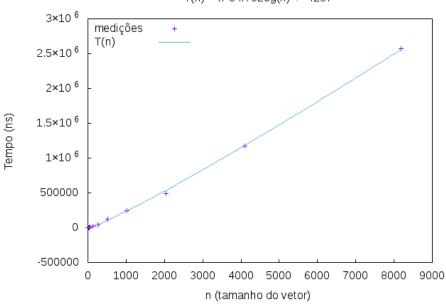


Figure 11: Heapsort vetor parcialmente decrescente 90 %

Método HeapSort com vetor parcialmente decrescente(90) T(n) = n\*33.38Log(n) + 1.441e+04

