Análise experimental de algoritmos usando Python

Karen Catiguá Junqueira
karen@ufu.com

Matheus Prado Prandini Faria
matheus_prandini@ufu.com

Pedro Augusto Correa Braz
pedro_acbraz@hotmail.com

Faculdade de Computação
Universidade Federal de Uberlândia

16 de dezembro de 2016

Lista de Figuras

Lista de Tabelas

Lista de Listagens

Sumário

Capítulo 1

Análise

O algoritmo selection sort é considerado um dos mais simples. Nele, temos um laço "for"aninhado a outro laço "for", os quais serão executados para qualquer vetor de entrada, de forma que o segundo laço seja verificado $n * (n - 1)$ vezes. Assim, a complexidade de tempo será $teta(n^2)$ tanto no melhor caso quanto no pior caso. Além disso, serão sempre realizadas $teta(n^2)$ comparações e um número linear de trocas, $teta(n)$. A complexidade de espaço é constante, $teta(1)$, pois é preciso espaço adicional somente para variáveis locais.

Capítulo 2

Resultados

2.1 Tabelas

n	comparações	tempo(s)
32	496	0.000674
64	2016	0.002528
128	8128	0.009982
256	32640	0.041368
512	130816	0.149926
1024	523776	0.602251
2048	2096128	2.589130
4096	8386560	9.723850
8192	33550336	38.927500

Tabela 2.1: *SelectionSort com Vetores Aleatorio*

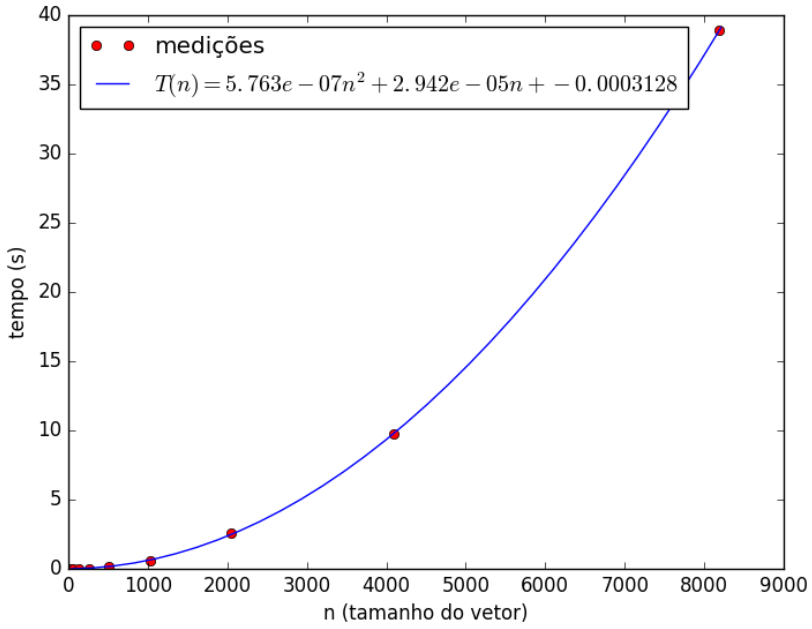

Figura 2.1: *Selection com relação ao tempo com vetor aleatório.*

Análise com relação ao tamanho do vetor 2^{32}:

$5.763 * 10^{-7} * n^2 + 2.942 * 10^{-5} * n - 0.0003128 = 1.063085874e + 13$

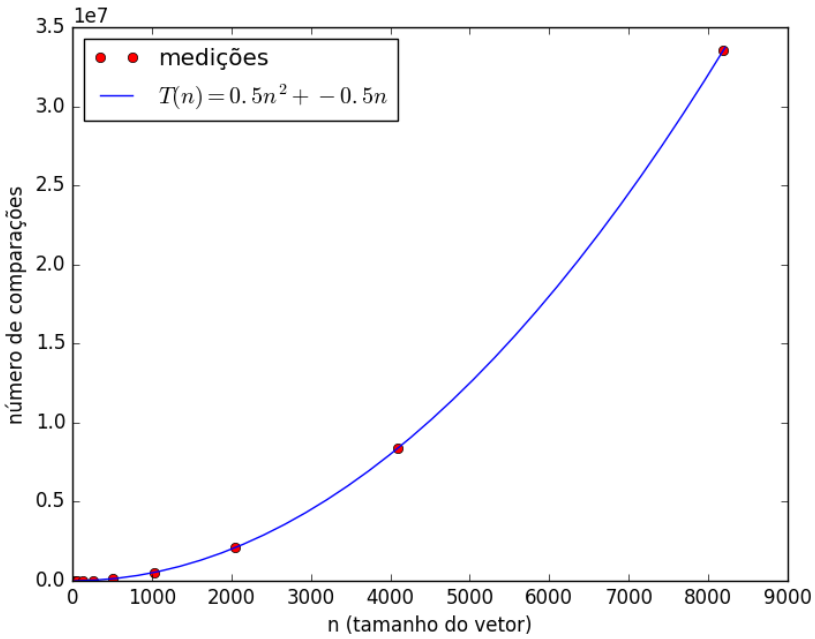

Figura 2.2: *Selection com relação ao número de comparações com vetor aleatório.*

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000730
64	2016	0.002492
128	8128	0.009608
256	32640	0.036564
512	130816	0.146064
1024	523776	0.620764
2048	2096128	2.523420
4096	8386560	9.821810
8192	33550336	34.955500

Tabela 2.2: *SelectionSort com Vetores Crescentes*

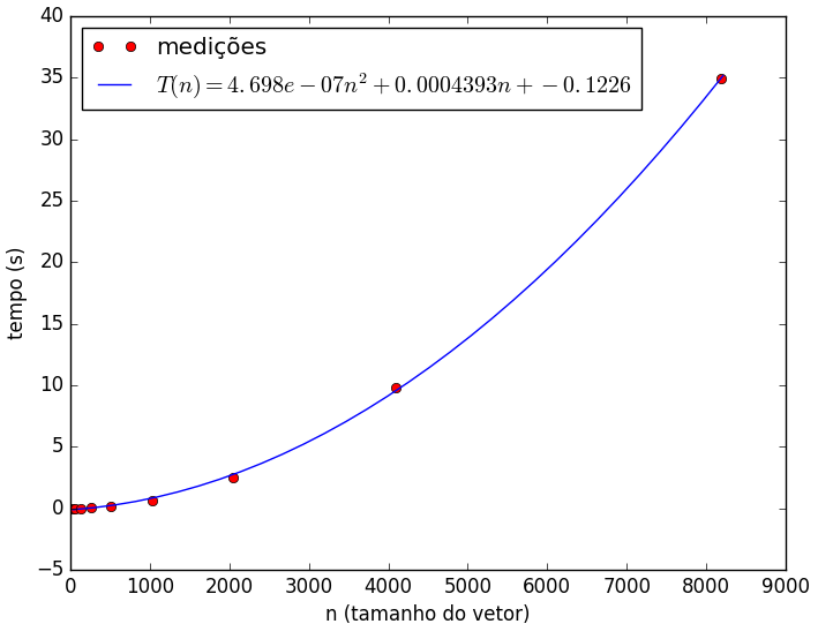

Figura 2.3: *Selection com relação ao tempo com vetor crescente.*

Análise com relação ao tamanho do vetor 2^{32}:

$4.698 * 10^{-7} * n^2 + 0.0004393 * n - 0.1226 = 8.66628e + 12$

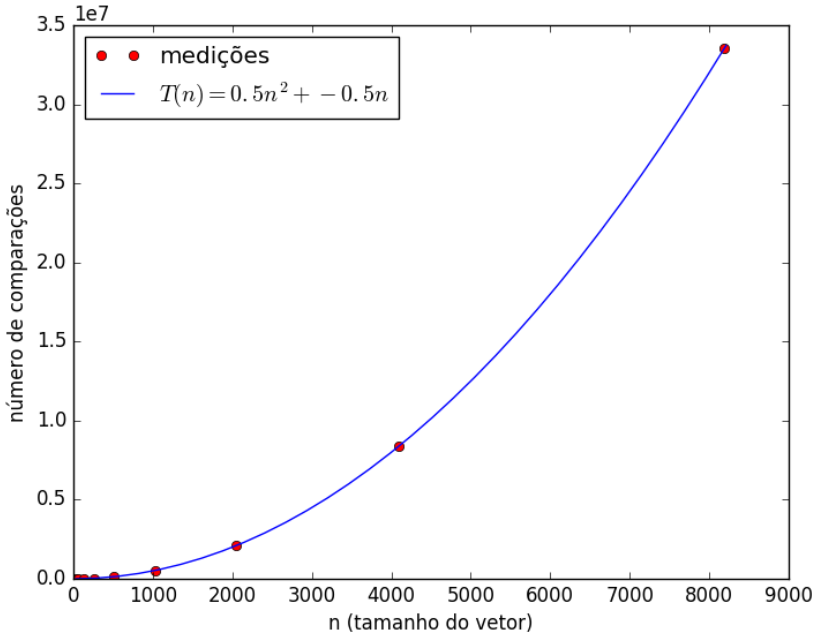

Figura 2.4: *Selection com relação ao número de comparações com vetor crescente.*

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000802
64	2016	0.002824
128	8128	0.011335
256	32640	0.042696
512	130816	0.182190
1024	523776	0.675674
2048	2096128	2.832220
4096	8386560	10.734100
8192	33550336	41.775800

Tabela 2.3: *SelectionSort com Vetores Decrescente*

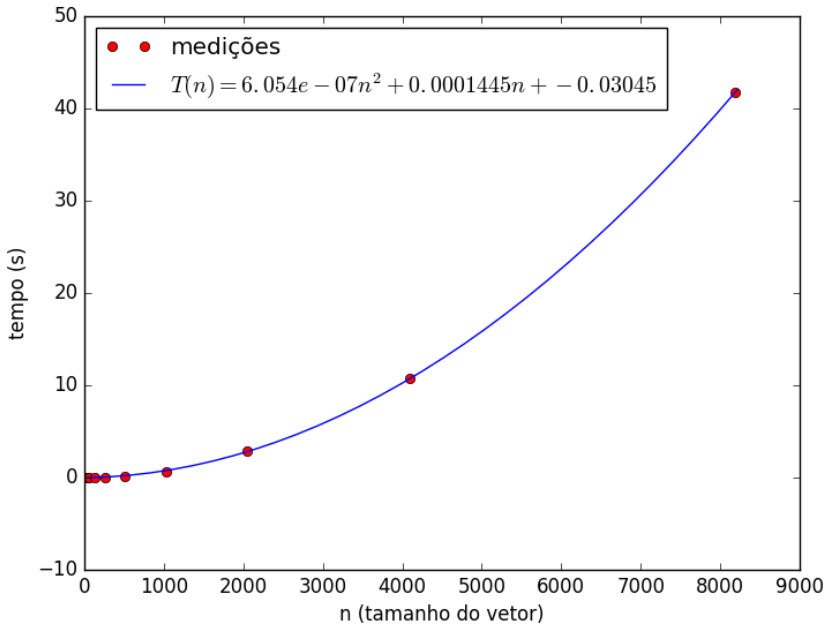

Figura 2.5: *Selection com relação ao tempo com vetor decrescente.*

Análise com relação ao tamanho do vetor 2^{32}:

$6.054 * 10^{-7} * n^2 + 0.0001445 * n - 0.03045 = 3.3502976e + 20$

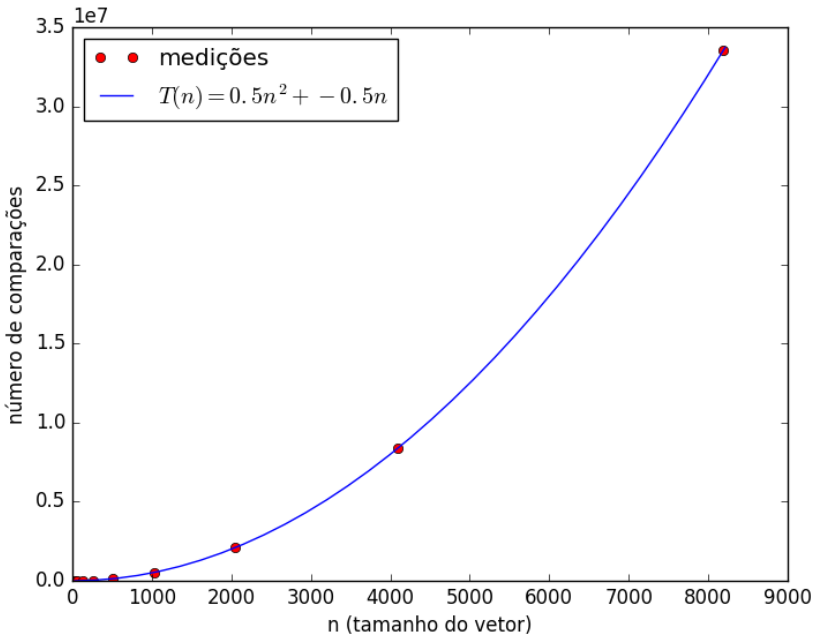

Figura 2.6: *Selection com relação ao número de comparações com vetor decrescente.*

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000795
64	2016	0.002461
128	8128	0.009831
256	32640	0.035777
512	130816	0.160543
1024	523776	0.601333
2048	2096128	2.240960
4096	8386560	8.678240
8192	33550336	36.259300

Tabela 2.4: *SelectionSort com Vetores Quase Crescentes* 10%

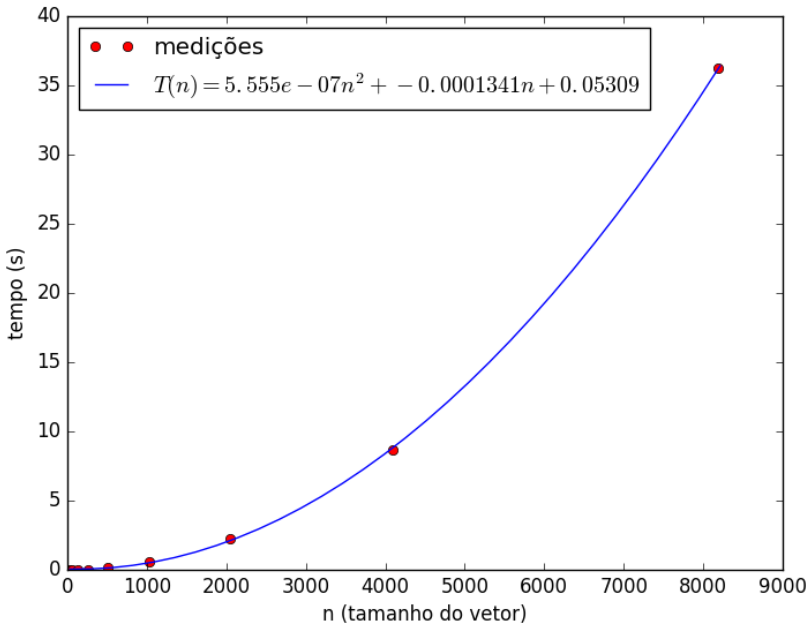

Figura 2.7: *selection com relação ao tempo com vetor quase crescente* 10%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.555 * 10^{-7} * n^2 + 0.0001341 * n + 0.0539 = 1.024716e + 13$

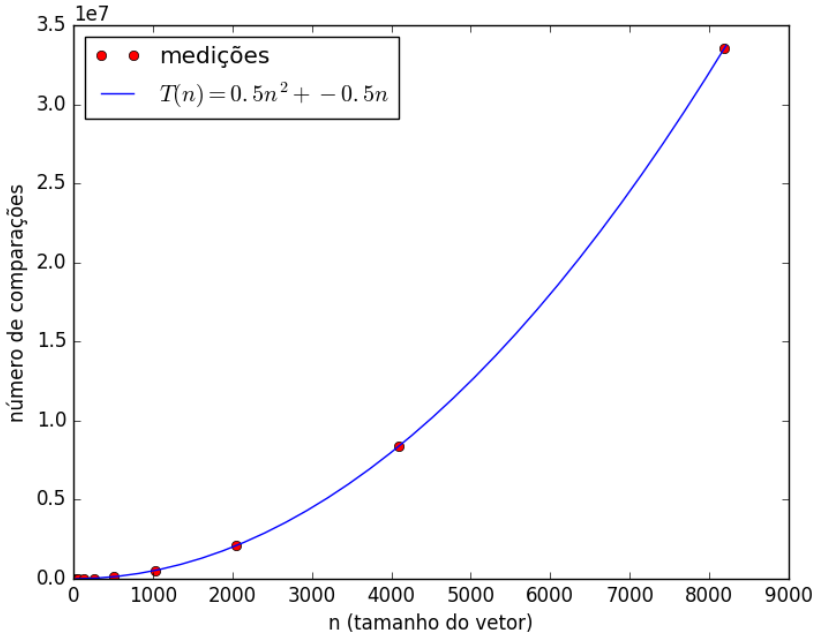

Figura 2.8: *Selection com relação ao número de comparações com vetor quase crescente* 10%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000651
64	2016	0.002602
128	8128	0.008712
256	32640	0.040318
512	130816	0.157252
1024	523776	0.538567
2048	2096128	2.258560
4096	8386560	10.138500
8192	33550336	39.937600

Tabela 2.5: *SelectionSort com Vetores Quase Crescentes* 20%

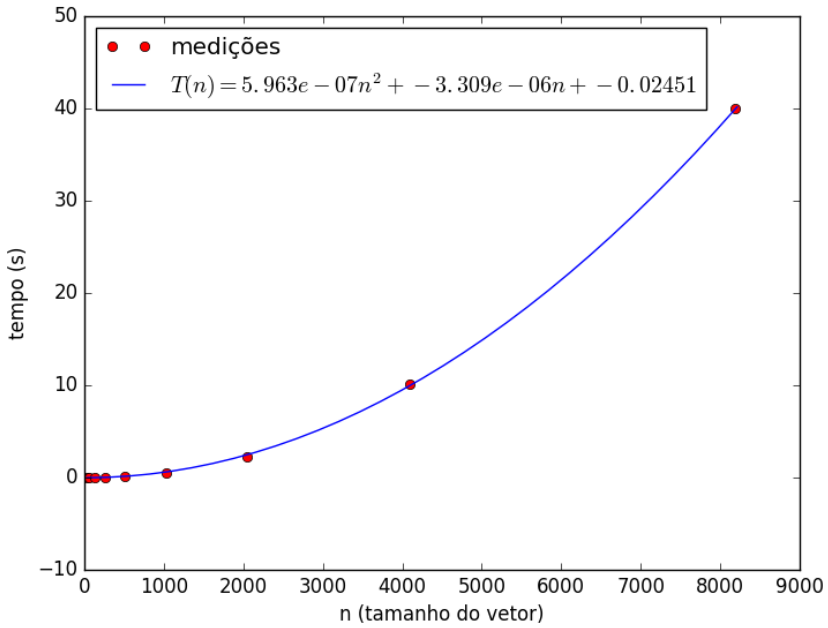

Figura 2.9: *Selection com relação ao tempo com vetor quase crescente* 20%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.963 * 10^{-7} * n^2 + 3.309 * 10^{-6} * n - 0.02451 = 1.099979e + 13$

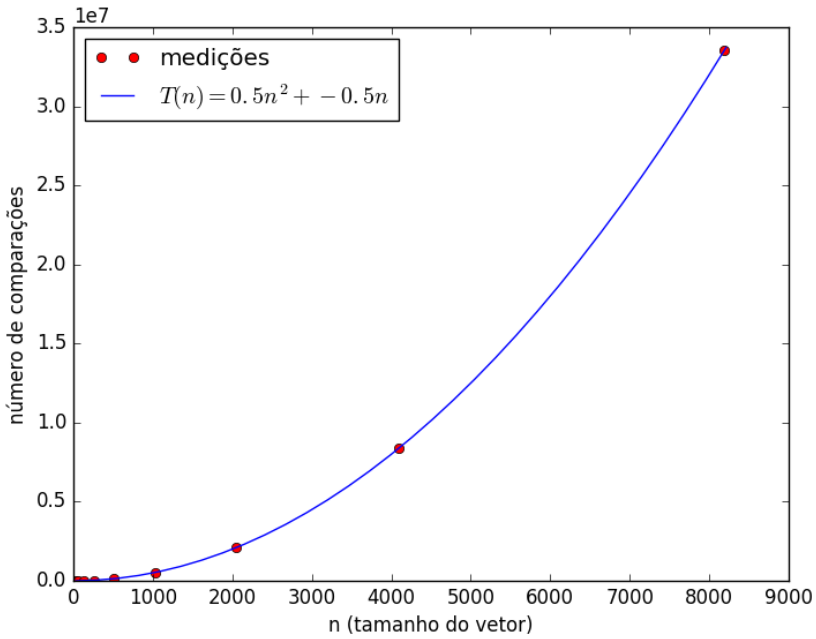

Figura 2.10: *Selection com relação ao número de comparações com vetor quase crescente* 20%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000702
64	2016	0.002577
128	8128	0.010371

Tabela 2.6: *SelectionSort com Vetores Quase Crescentes* 30%

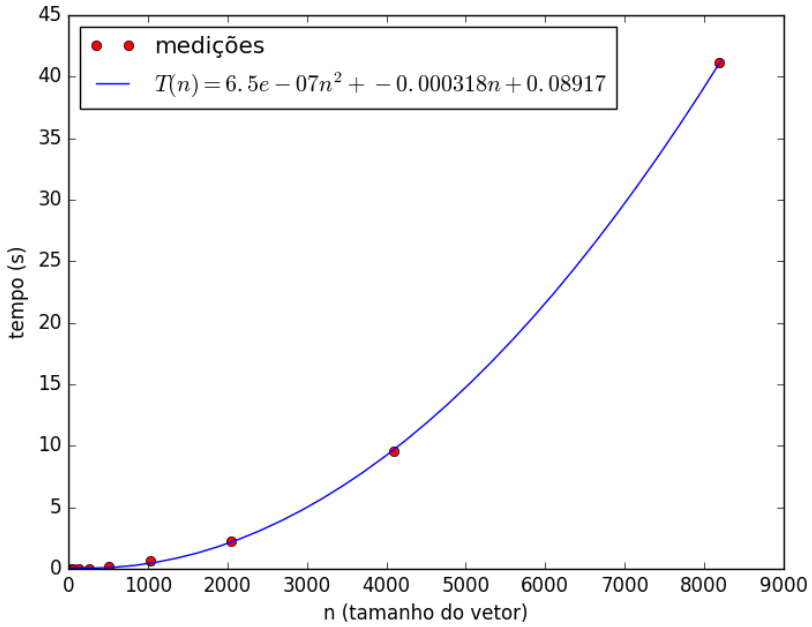

Figura 2.11: *Selection com relação ao tempo com vetor quase crescente* 30%.

Análise com relação ao tamanho do vetor 2^{32}:

$6.5 * 10^{-7} * n^2 - 0.000318 * n + 0.08917 = 1.199038e + 13$

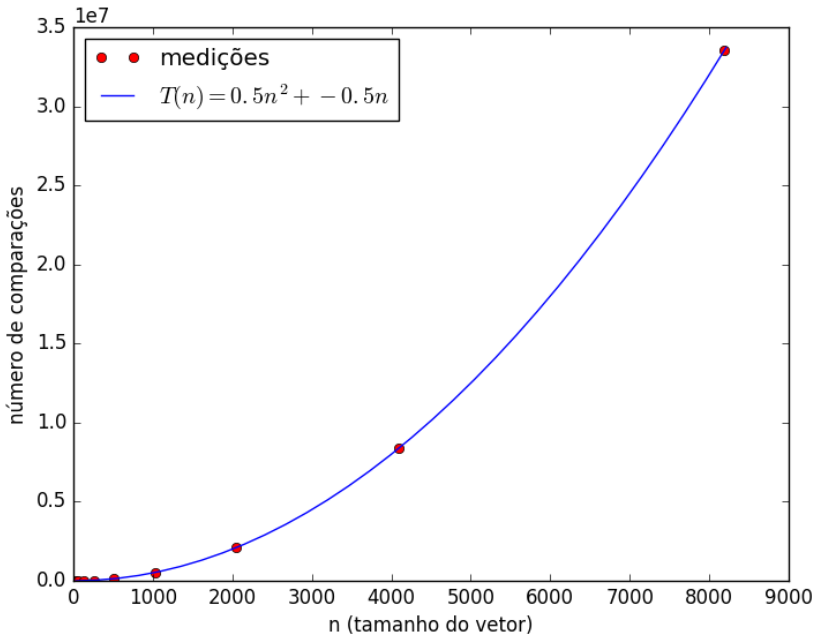

Figura 2.12: *Selection com relação ao número de comparações com vetor quase crescente* 30%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000678
64	2016	0.002848
128	8128	0.010051
256	32640	0.036314
512	130816	0.152581
1024	523776	0.571454
2048	2096128	2.427220
4096	8386560	10.092500
8192	33550336	38.888700

Tabela 2.7: *SelectionSort com Vetores Quase Crescentes* 40%

2.1

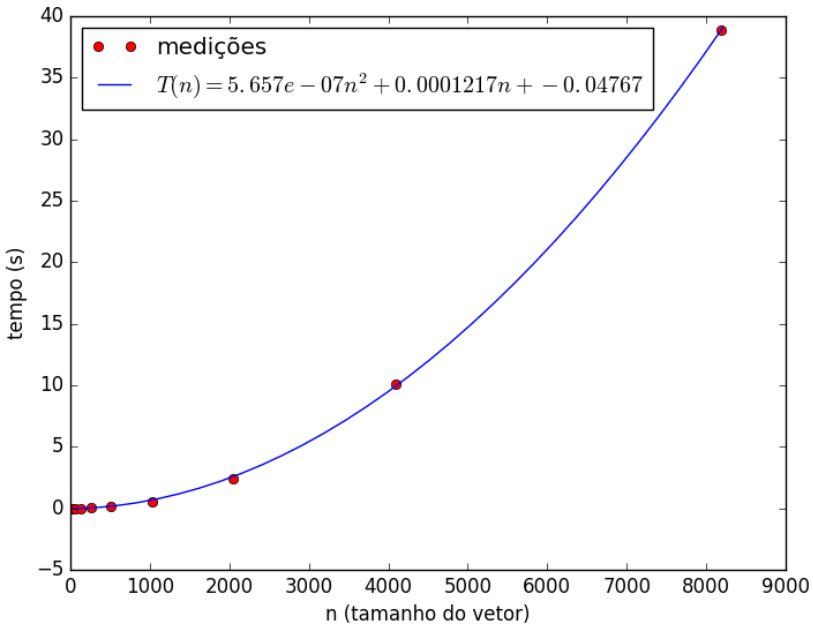

Figura 2.13: *Selection com relação ao tempo com vetor quase crescente* 40%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.657 * 10^{-7} * n^2 + 0.0001217 * n - 0.04767 = 1.043532e + 13$

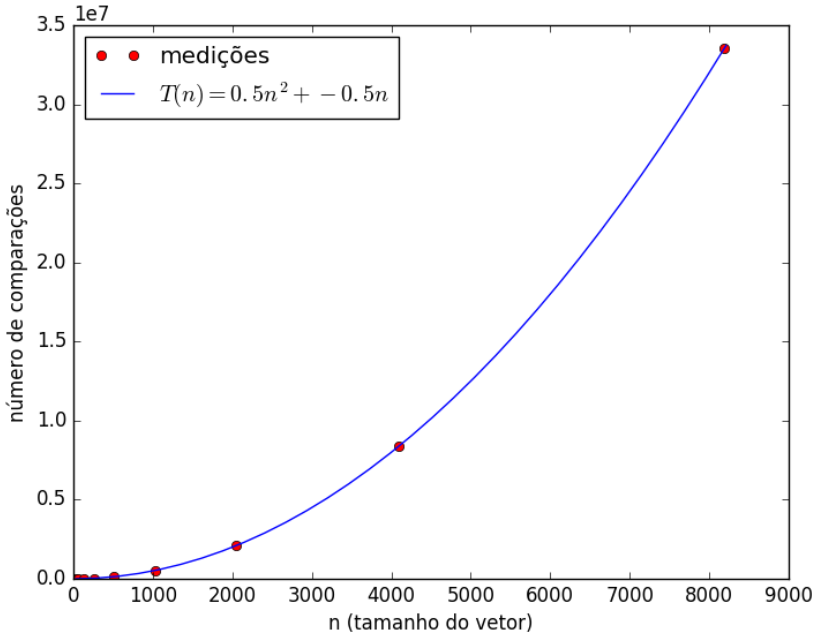

Figura 2.14: *Selection com relação ao número de comparações com vetor quase crescente* 40%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000744
64	2016	0.002493
128	8128	0.009576
256	32640	0.034961
512	130816	0.156885
1024	523776	0.568045
2048	2096128	2.502620
4096	8386560	10.183400
8192	33550336	37.891200

Tabela 2.8: *SelectionSort com Vetores Quase Crescentes* 50%

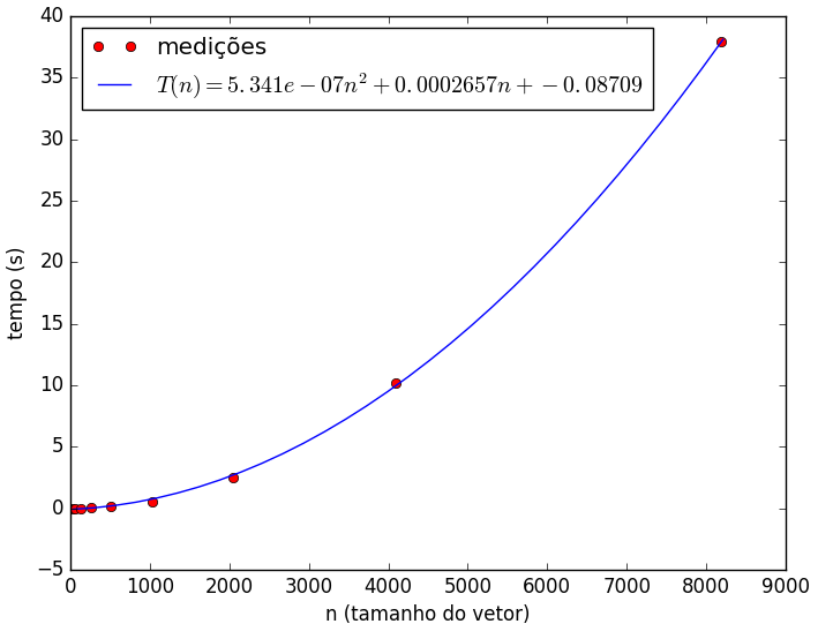

Figura 2.15: *Selection com relação ao tempo com vetor quase crescente* 50%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.341 * 10^{-7} * n^2 + 0.0002657 * n - 0.08709 = 9.852407e + 12$

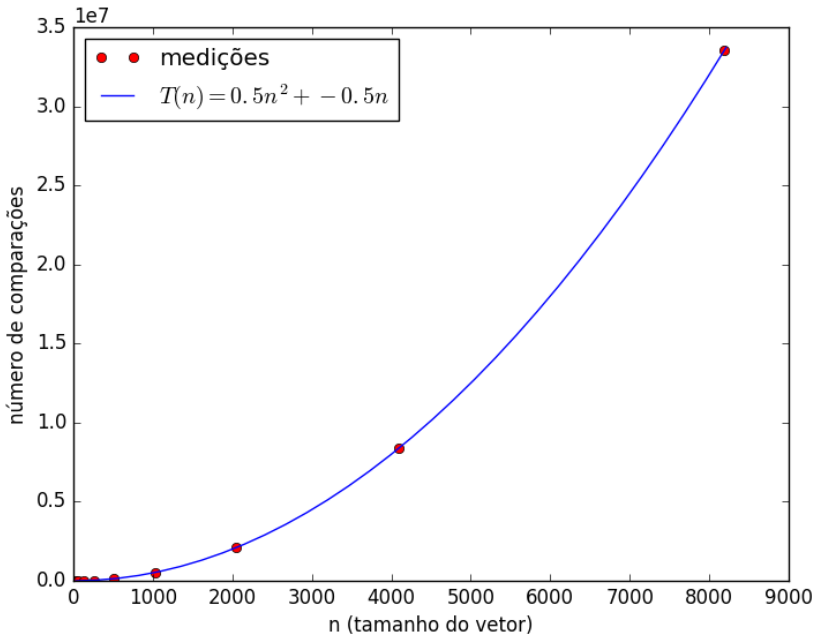

Figura 2.16: *Selection com relação ao número de comparações com vetor quase crescente* 50%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000831
64	2016	0.002516
128	8128	0.011099
256	32640	0.049164
512	130816	0.184920
1024	523776	0.643137
2048	2096128	2.729810
4096	8386560	10.310400
8192	33550336	39.486300

Tabela 2.9: *SelectionSort com Vetores Quase Decrescentes* 10%

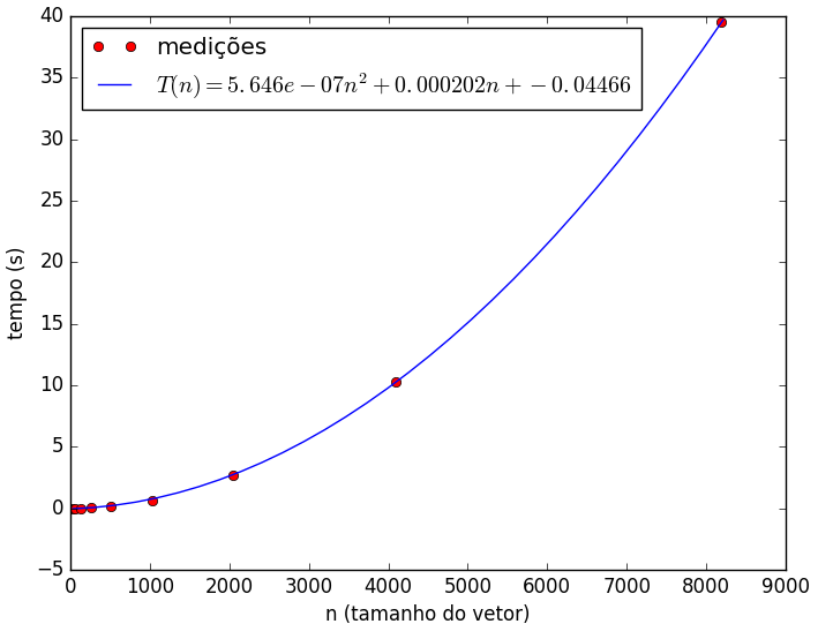

Figura 2.17: *Selection com relação ao tempo com vetor quase decrescente* 10%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.646 * 10^{-7} * n^2 - 0.000202 * n - 0.04466 = 1.041503e + 13$

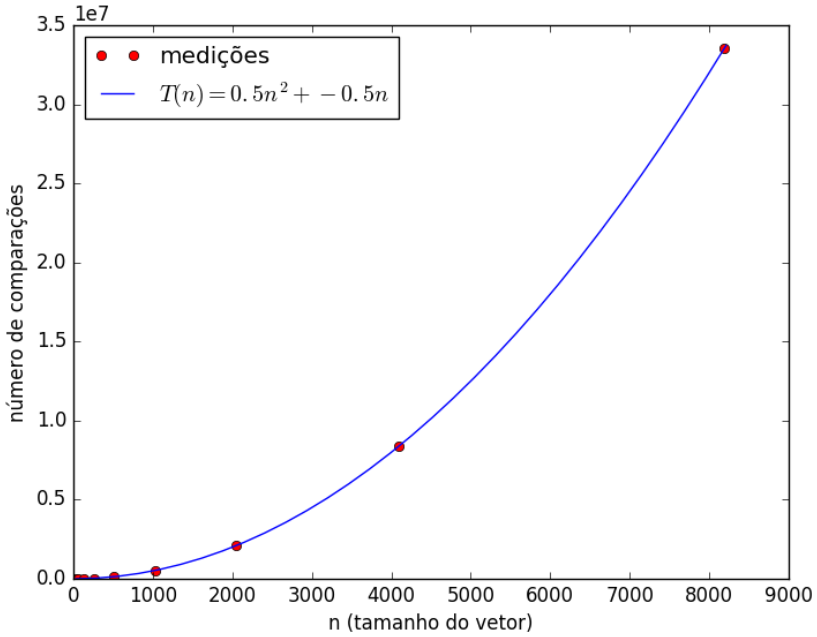

Figura 2.18: *Selection com relação ao número de comparações com vetor quase decrescente* 10%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000763
64	2016	0.002921
128	8128	0.010516

Tabela 2.10: *SelectionSort com Vetores Quase Decrescentes* 20%

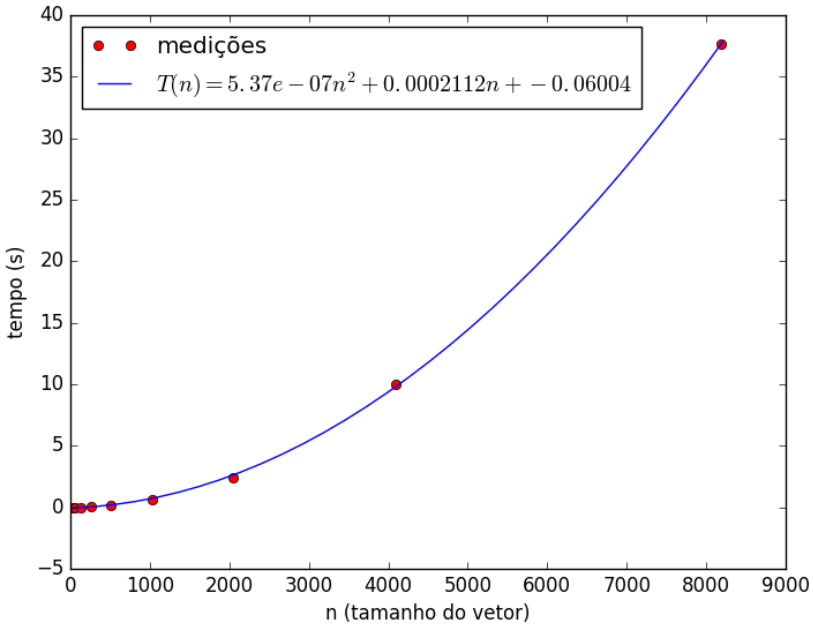

Figura 2.19: *Selection com relação ao tempo com vetor quase decrescente* 20%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.37 * 10^{-7} * n^2 - 0.0002112 * n - 0.06004 = 9.9059006e + 12$

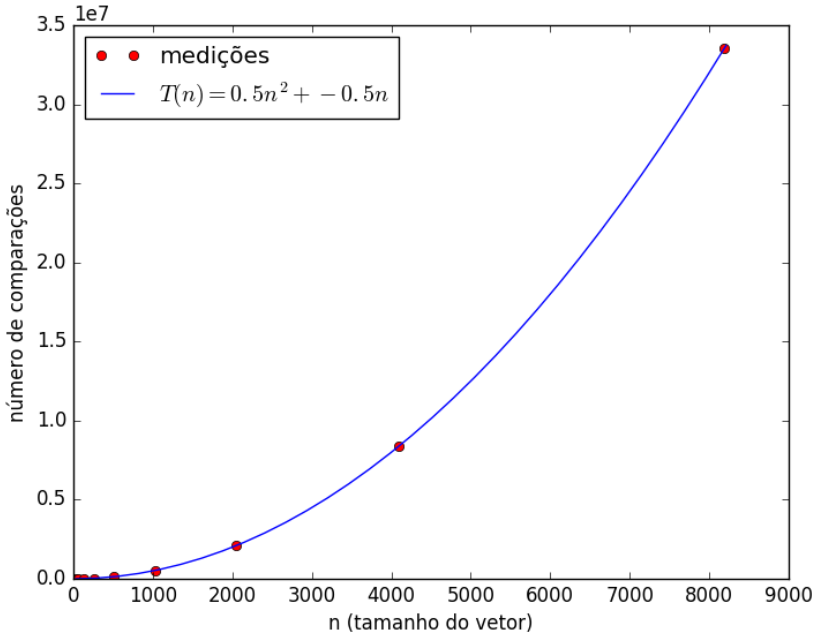

Figura 2.20: *Selection com relação ao número de comparações com vetor quase decrescente* 20%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000650
64	2016	0.002867
128	8128	0.010118

Tabela 2.11: *SelectionSort com Vetores Quase Decrescentes* 30%

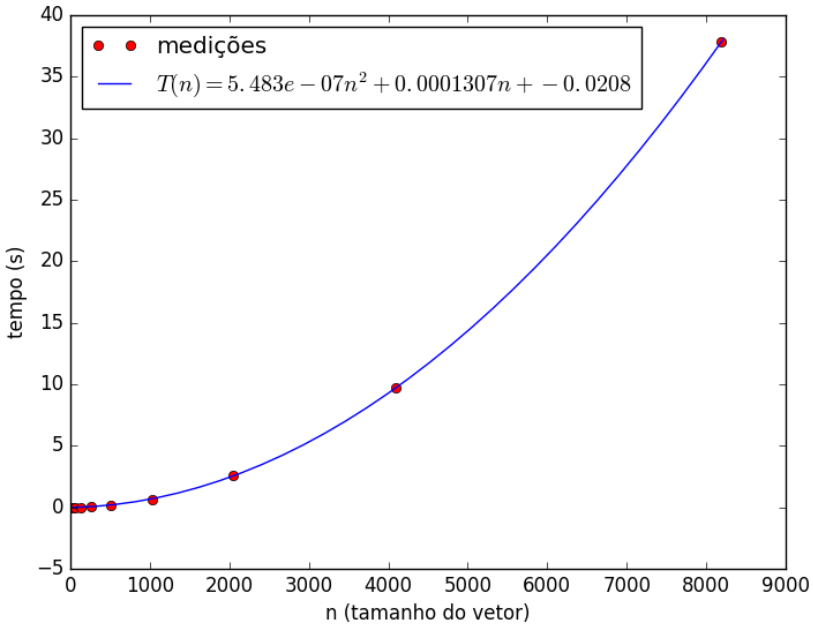

Figura 2.21: *Selection com relação ao tempo com vetor quase decrescente* 30%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.482 * 10^{-7} * n^2 - 0.0001307 * n - 0.0208 = 1.0112504e + 13$

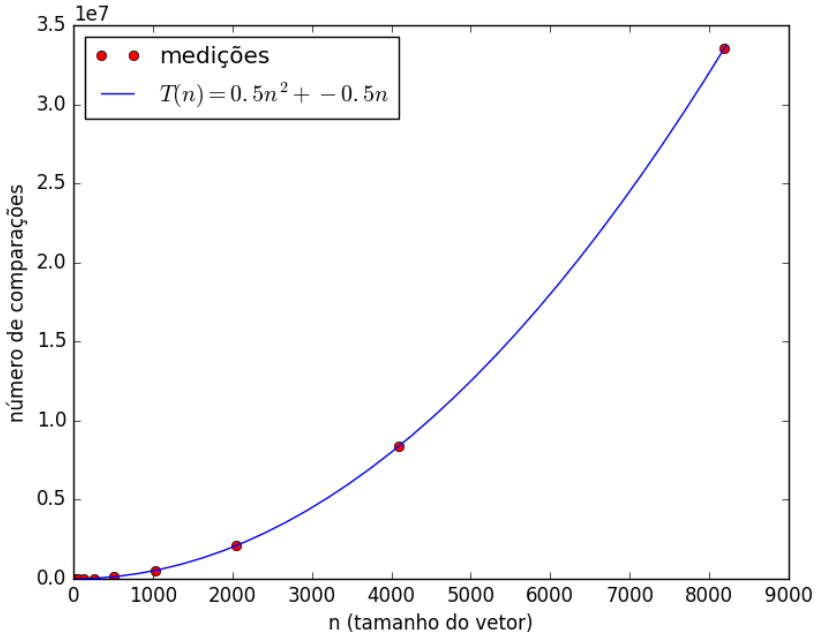

Figura 2.22: *Selection com relação ao número de comparações com vetor quase decrescente* 30%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000793
64	2016	0.002781
128	8128	0.010400

Tabela 2.12: *SelectionSort com Vetores Quase Decrescentes* 40%

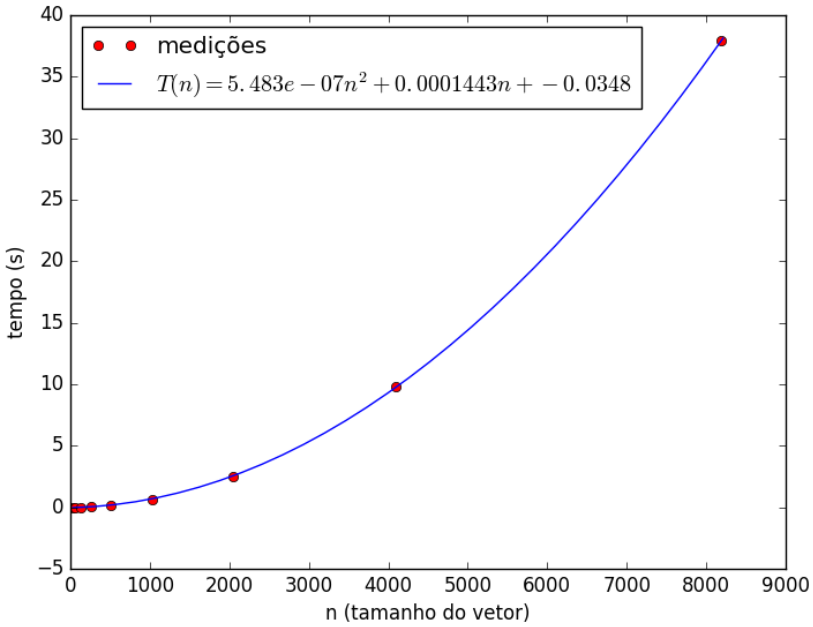

Figura 2.23: *Selection com relação ao tempo com vetor quase decrescente* 40%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.483 * 10^{-7} * n^2 - 0.0001443 * n - 0.0348 = 1.0114349e + 13$

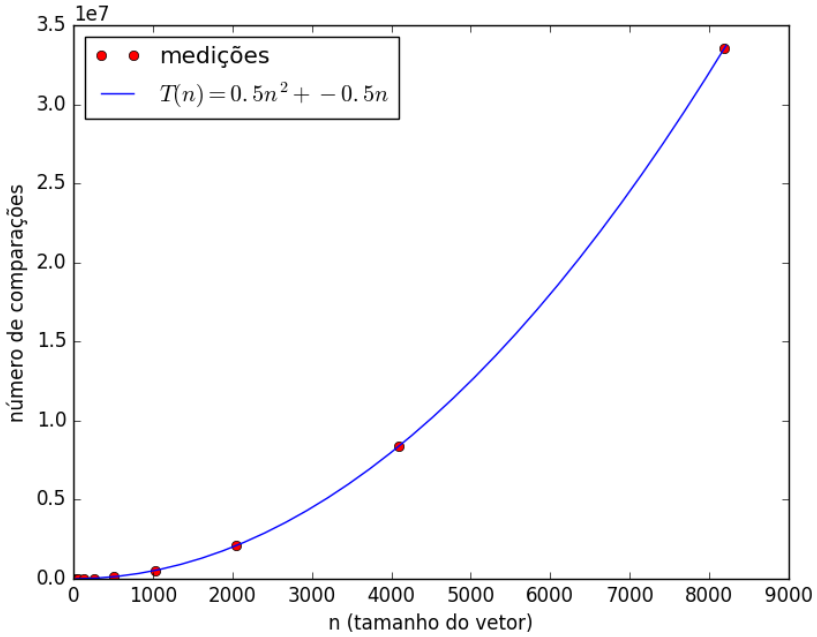

Figura 2.24: *Selection com relação ao número de comparações com vetor quase decrescente* 40%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

n	comparações	tempo(s)
32	496	0.000724
64	2016	0.003108
128	8128	0.009988
256	32640	0.042663
512	130816	0.166300
1024	523776	0.711966
2048	2096128	2.696660
4096	8386560	10.460000
8192	33550336	39.957700

Tabela 2.13: *SelectionSort com Vetores Quase Decrescentes* 50%

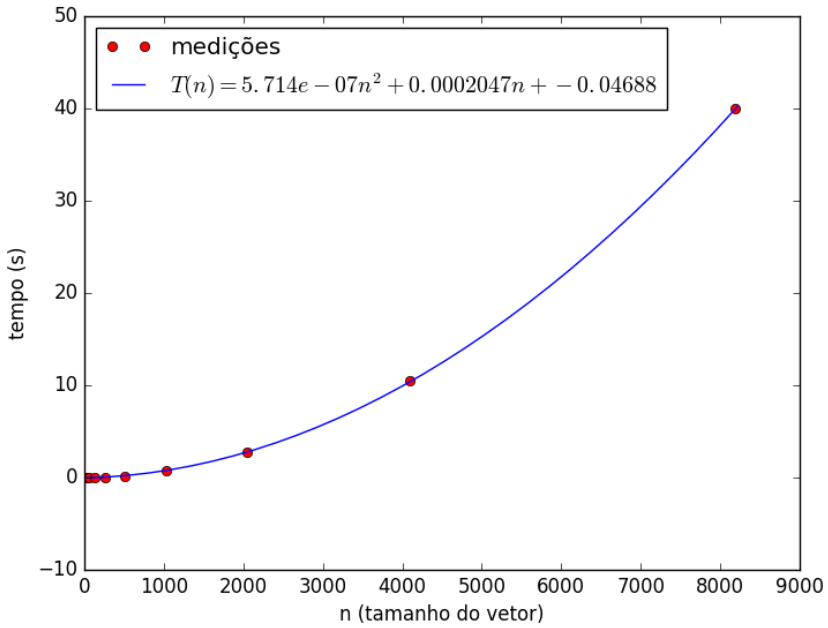

Figura 2.25: *Selection com relação ao tempo com vetor quase decrescente* 50%.

Análise com relação ao tamanho do vetor 2^{32}:

$5.714 * 10^{-7} * n^2 + 0.0002047 * n - 0.04688 = 1.0540470e + 13$

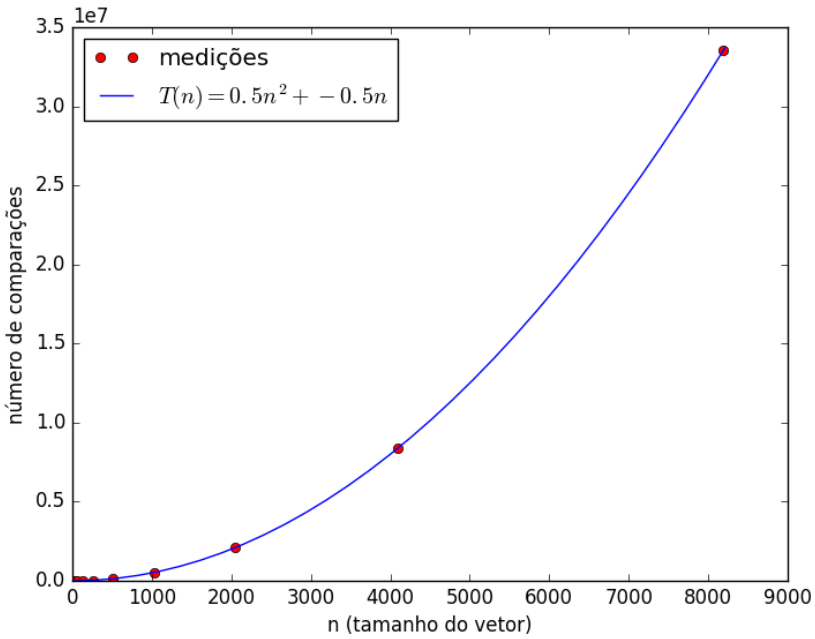

Figura 2.26: *Selection com relação ao número de comparações com vetor quase decrescente* 50%.

Análise com relação ao tamanho do vetor 2^{32}:

$0.5 * n^2 - 0.5 * n = 9.223372e + 18$

Apêndice A

Arquivo ../selection/selection.py

Listagem A.1: ../selection/selection.py

```
1  import sys
2
3
4
5  #from memoria import *
6
7  @profile
8  def selectionSort(A):
9    for i in range(0,(len(A)-1)):
10     minimo = i
11     for j in range(i+1, len(A)):
12       if A[j] < A[minimo]:
13         minimo = j
14     aux = A[i]
15     A[i] = A[minimo]
16     A[minimo] = aux
17
18
19 #A = [40, 12, 34, 1, 3, 5, 80]
20 #selectionSort(A)
21 #print(A)
```

Apêndice B

Arquivo ../selection/ensaio.py

Listagem B.1: ../selection/ensaio.py

```
1  import numpy as np
2  import argparse
3
4  from selectionsort import *
5
6  parser = argparse.ArgumentParser()
7  parser.add_argument("arq_vetor",
8                      help="nome do arquivo contendo o vetor de teste")
9  args = parser.parse_args()
10
11 # Lê o arquivo contendo o vetor e passado na linha de comando como um
12 # vetor do Numpy.
13
14 vet = np.loadtxt(args.arq_vetor)
15 selectionSort(vet)
```