

Análise experimental de algoritmos usando Python

Patrícia Mariana Ramos Marcolino

`pmrmarcolino@hotmail.com`

Eduardo Pinheiro Barbosa

`eduardptu@hotmail.com`

Faculdade de Computação
Universidade Federal de Uberlândia

1 de julho de 2016

Lista de Figuras

2.1	A análise do grafico para 2^{32} segue abaixo para selectionsort	8
2.2	A análise do grafico para 2^{32} segue abaixo para selectionsort	9
2.3	A análise do grafico para 2^{32} segue abaixo para selectionsort	10
2.4	A análise do grafico para 2^{32} segue abaixo para selectionsort	11
2.5	A análise do grafico para 2^{32} segue abaixo para selectionsort	12
2.6	A análise do grafico para 2^{32} segue abaixo para selectionsort	13
2.7	A análise do grafico para 2^{32} segue abaixo para selectionsort	14
2.8	A análise do grafico para 2^{32} segue abaixo para selectionsort	15
2.9	A análise do grafico para 2^{32} segue abaixo para selectionsort	16
2.10	A análise do grafico para 2^{32} segue abaixo para selectionsort	17
2.11	A análise do grafico para 2^{32} segue abaixo para selectionsort	19
2.12	A análise do grafico para 2^{32} segue abaixo para selectionsort	20
2.13	A análise do grafico para 2^{32} segue abaixo para selectionsort	21
2.14	A análise do grafico para 2^{32} segue abaixo para selectionsort	22
2.15	A análise do grafico para 2^{32} segue abaixo para selectionsort	23
2.16	A análise do grafico para 2^{32} segue abaixo para selectionsort	24
2.17	A análise do grafico para 2^{32} segue abaixo para selectionsort	25
2.18	A análise do grafico para 2^{32} segue abaixo para selectionsort	27
2.19	A análise do grafico para 2^{32} segue abaixo para selectionsort	28
2.20	A análise do grafico para 2^{32} segue abaixo para selectionsort	30
2.21	A análise do grafico para 2^{32} segue abaixo para selectionsort	31
2.22	A análise do grafico para 2^{32} segue abaixo para selectionsort	32
2.23	A análise do grafico para 2^{32} segue abaixo para selectionsort	33
2.24	A análise do grafico para 2^{32} segue abaixo para selectionsort	34
2.25	A análise do grafico para 2^{32} segue abaixo para selectionsort	35

Lista de Tabelas

2.1	Tabela com vetor teste aleatório: A linha te interesse analisada para este caso é a 15.	7
2.2	Tabela com vetor teste crescente: A linha te interesse analisada para este caso é a 15.	7
2.3	Tabela com vetor teste decrescente: A linha te interesse analisada para este caso é a 15.	8
2.4	Tabela com vetor teste quase crescente 10%: A linha te interesse analisada para este caso é a 15.	9
2.5	Tabela com vetor teste quase crescente 20%: A linha te interesse analisada para este caso é a 15.	10
2.6	Tabela com vetor teste quase crescente 30%: A linha te interesse analisada para este caso é a 15.	18
2.7	Tabela com vetor teste quase crescente 40%: A linha te interesse analisada para este caso é a 15.	18
2.8	Tabela com vetor teste quase crescente 50%: A linha te interesse analisada para este caso é a 15.	19
2.9	Tabela com vetor teste quase decrescente 10%: A linha te interesse analisada para este caso é a 15.	20
2.10	Tabela com vetor teste quase decrescente 20%: A linha te interesse analisada para este caso é a 15	21
2.11	Tabela com vetor teste quase decrescente 30%: A linha te interesse analisada para este caso é a 15	29
2.12	Tabela com vetor teste quase decrescente 40%: A linha te interesse analisada para este caso é a 15	29
2.13	Tabela com vetor teste quase decrescente 50%: A linha te interesse analisada para este caso é a 15	30

Lista de Listagens

A.1	../selectionsort/selectionsort.py	36
B.1	../selectionsort/ensaio.py	37

Sumário

Lista de Figuras	2
Lista de Tabelas	3
1 Análise	6
2 Resultados	7
2.1 Tabelas	7
 Apêndice	 36
A Arquivo ../selectionsort/selectionsort.py	36
B Arquivo ../selectionsort/ensaio.py	37

Capítulo 1

Análise

O número de comparações é dado pela recorrência:

$$T(n) = \begin{cases} 0 & \text{se } n = 1 \\ T(n-1) + n - 1 & \text{caso contrario} \end{cases} \quad (1.1)$$

Portanto, $\theta(n^2)$ comparações são executadas no pior caso.

Já o número de trocas é dado pela recorrência:

$$T(n) = \begin{cases} 0 & \text{se } n = 1 \\ T(n-1) + 1 & \text{caso contrario} \end{cases} \quad (1.2)$$

Portanto, $\theta(n)$ trocas são executadas no pior caso.

Capítulo 2

Resultados

2.1 Tabelas

n	comparações	tempo(s)
32	496	0.000617
64	2016	0.002415
128	8128	0.008469
256	32640	0.035583
512	130816	0.136828
1024	523776	0.533262
2048	2096128	2.061350
4096	8386560	8.589490
8192	33550336	34.127500

Tabela 2.1: Tabela com vetor teste aleatório: A linha de interesse analisada para este caso é a 15.

n	comparações	tempo(s)
32	496	0.000648
64	2016	0.002277
128	8128	0.008596
256	32640	0.033311
512	130816	0.139917
1024	523776	0.522387
2048	2096128	1.987380
4096	8386560	8.685150
8192	33550336	34.369400

Tabela 2.2: Tabela com vetor teste crescente: A linha de interesse analisada para este caso é a 15.

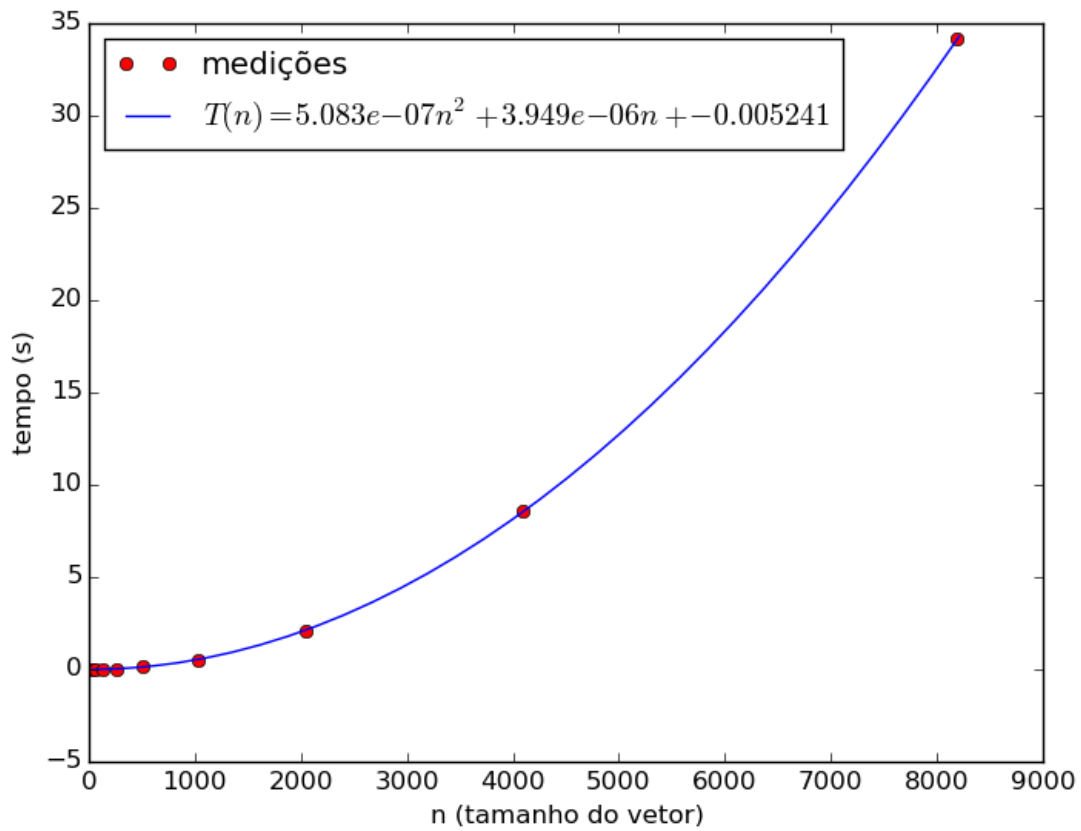


Figura 2.1: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 5.083e - 07 * n^2 + 3.949e - 06 * n - 0.0052$ e para o $n = 2^{32}$,
 $T(2^{32}) = 9.1376742 * 10^{301}$

n	comparações	tempo(s)
32	496	0.000672
64	2016	0.002618
128	8128	0.010014
256	32640	0.037622
512	130816	0.150541
1024	523776	0.604606
2048	2096128	2.596550
4096	8386560	9.657080
8192	33550336	34.009200

Tabela 2.3: Tabela com vetor teste decrescente: A linha de interesse analisada para este caso é a 15.

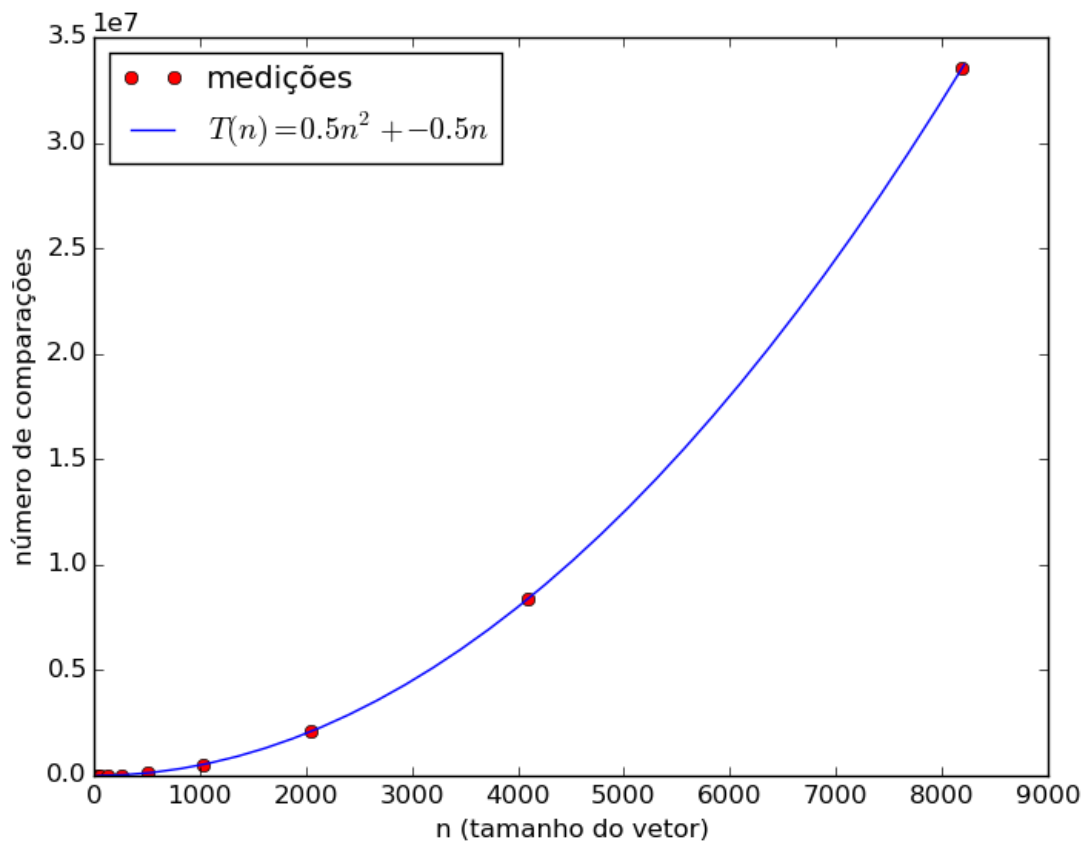


Figura 2.2: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000613
64	2016	0.002308
128	8128	0.008188
256	32640	0.033693
512	130816	0.129137
1024	523776	0.579641
2048	2096128	1.999810
4096	8386560	9.121400
8192	33550336	32.989500

Tabela 2.4: Tabela com vetor teste quase crescente 10%: A linha de interesse analisada para este caso é a 15.

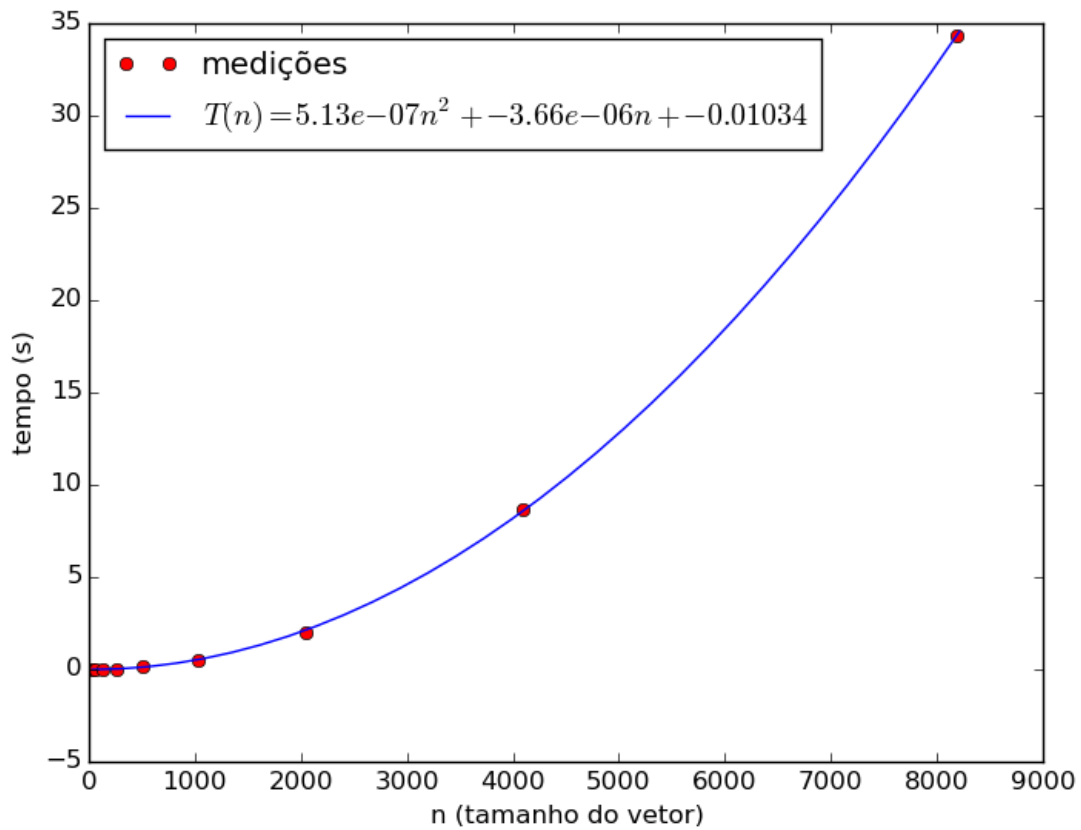


Figura 2.3: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 5.13e - 07 * n^2 + 3.66e - 06 * n - 0.001034$ e para o $n = 2^{32}$,
 $T(2^{32}) = 9.222165 * 10^{301}$

n	comparações	tempo(s)
32	496	0.000627
64	2016	0.002248
128	8128	0.009107
256	32640	0.034544
512	130816	0.128516
1024	523776	0.526562
2048	2096128	2.038870
4096	8386560	9.178640
8192	33550336	32.969100

Tabela 2.5: Tabela com vetor teste quase crescente 20%: A linha de interesse analisada para este caso é a 15.

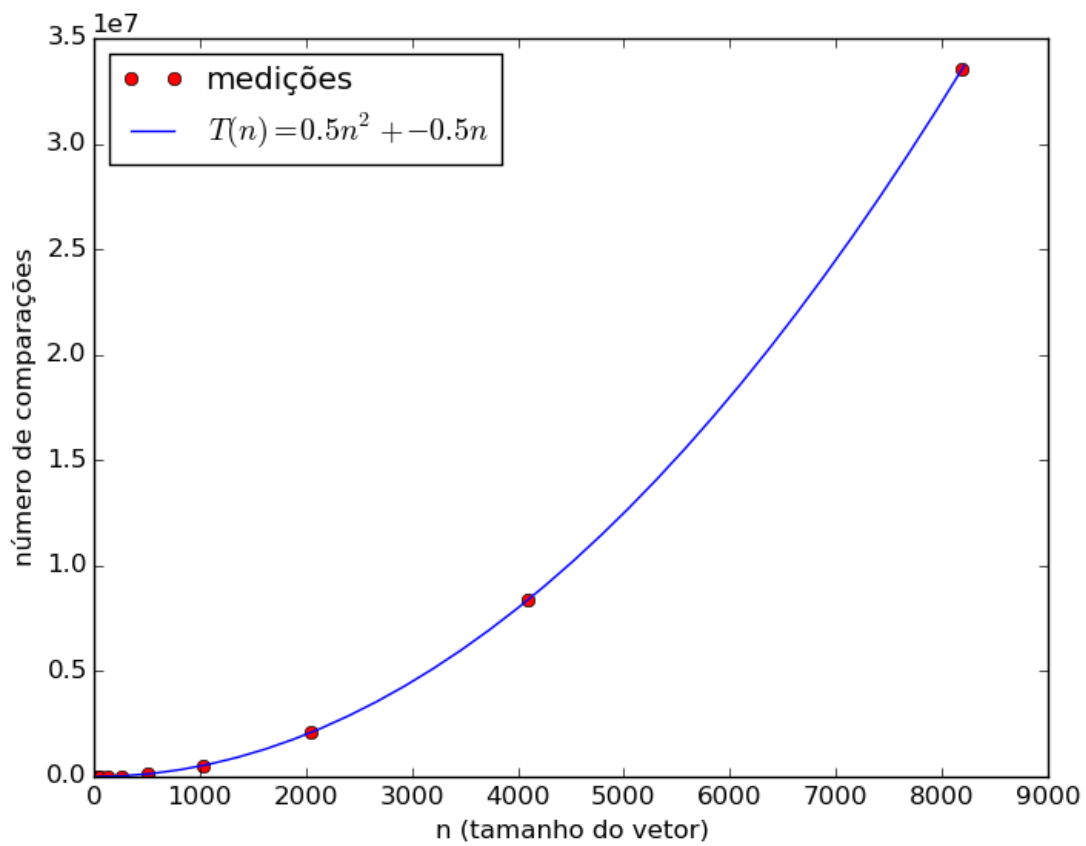


Figura 2.4: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

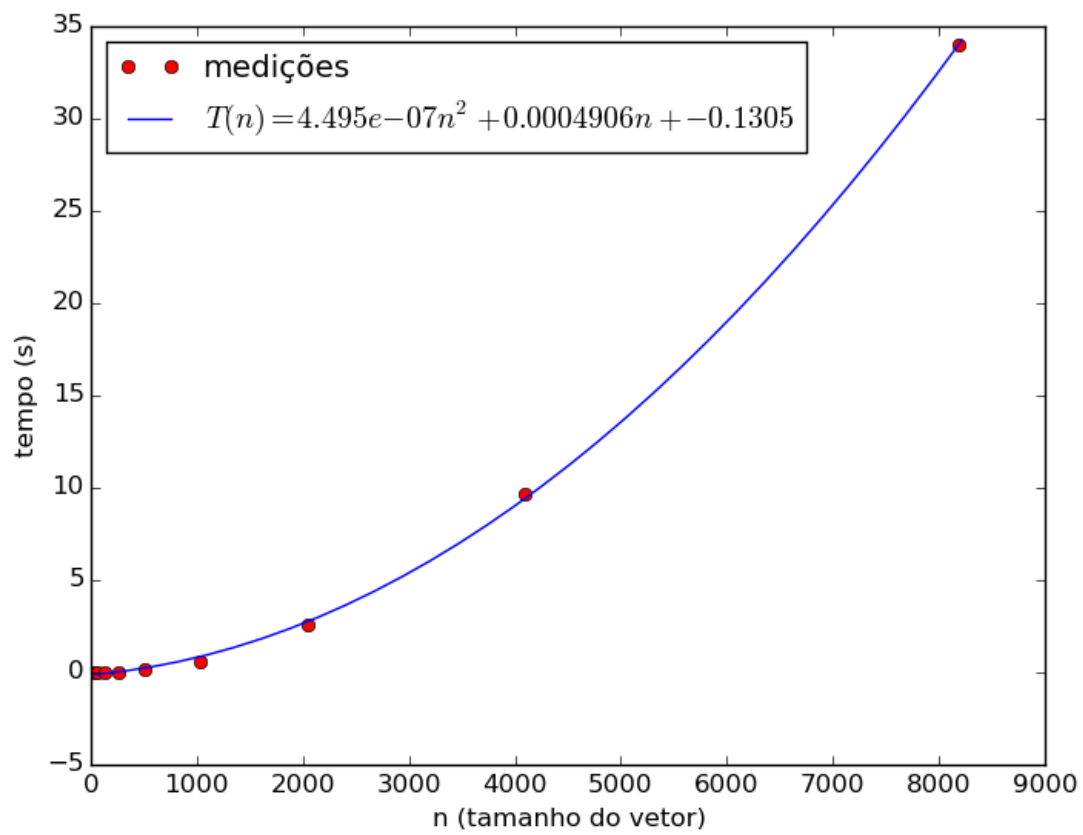


Figura 2.5: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.495e-07 * n^2 + 0.00049 * n + 0.1305$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.0806306 * 10^{301}$

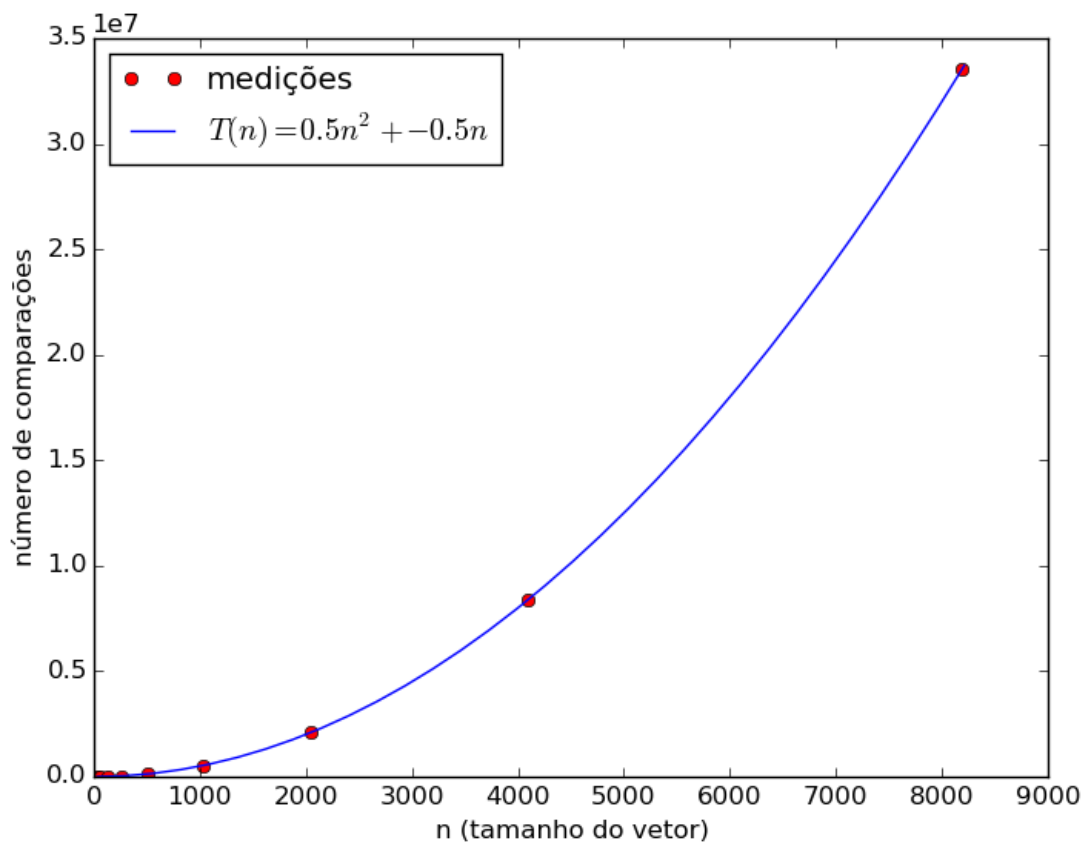


Figura 2.6: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

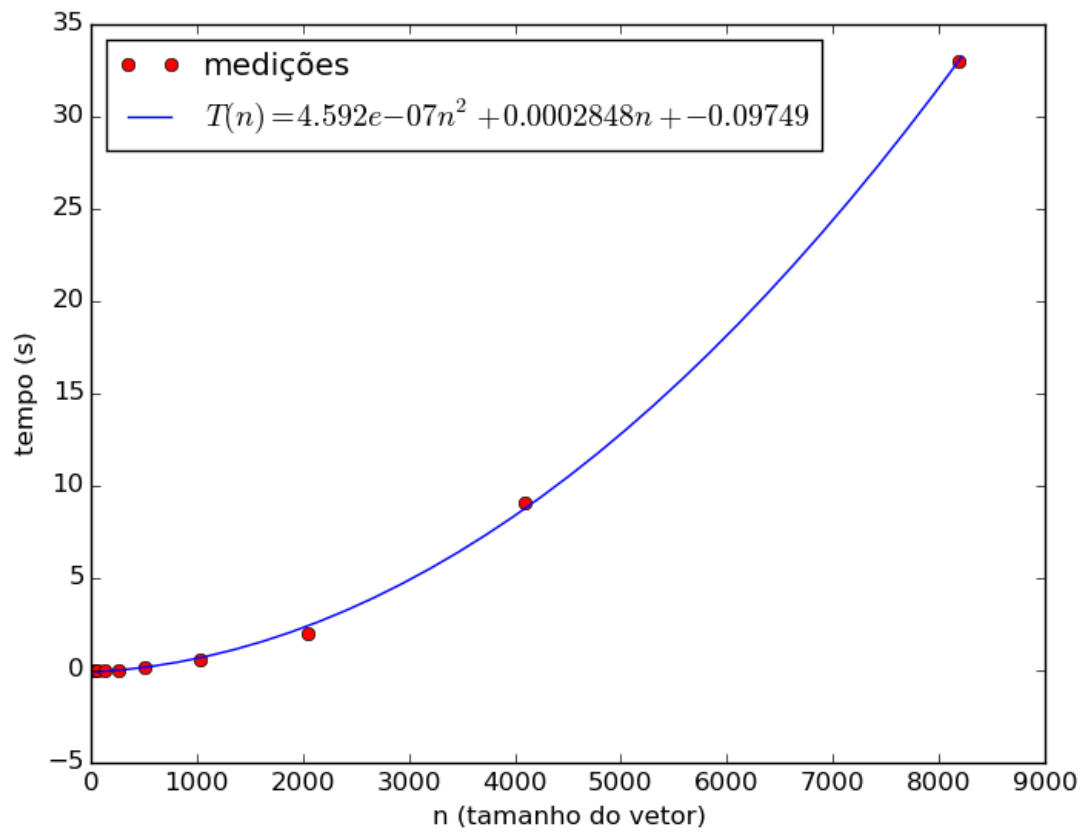


Figura 2.7: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.592e - 07 * n^2 + 0.000284 * n - 0.09749$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.25501 * 10^{301}$

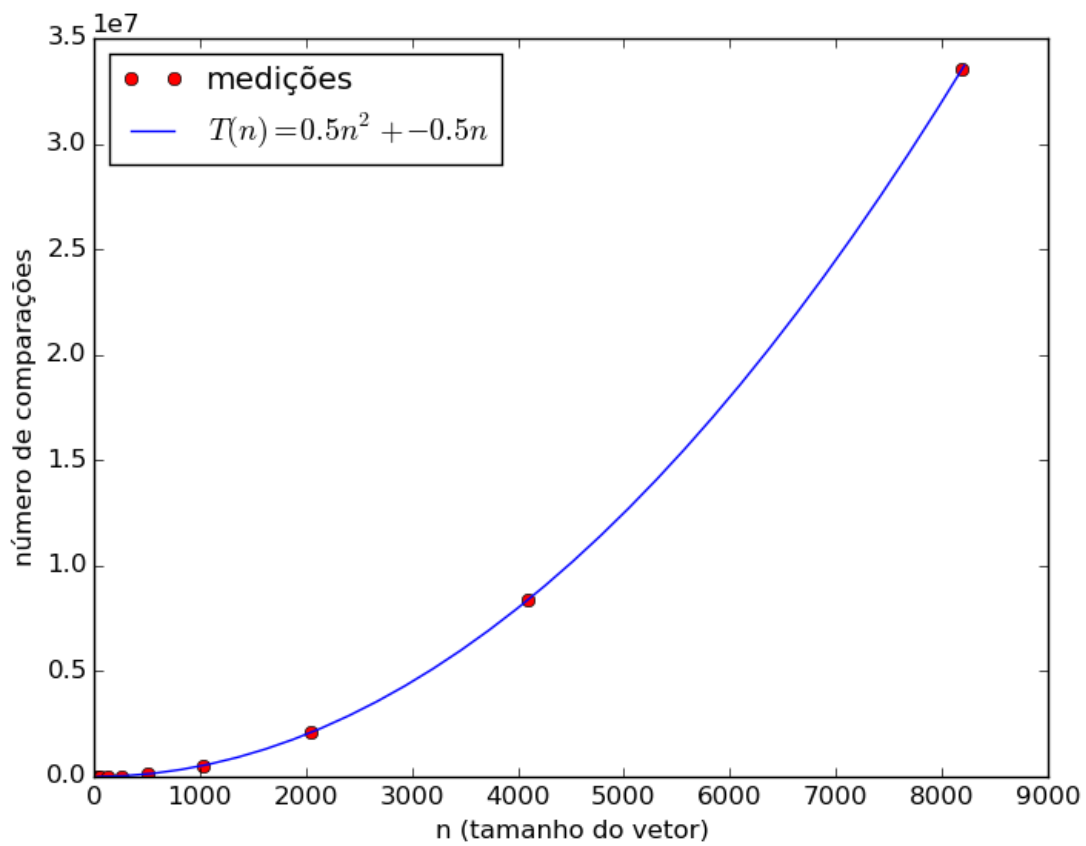


Figura 2.8: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

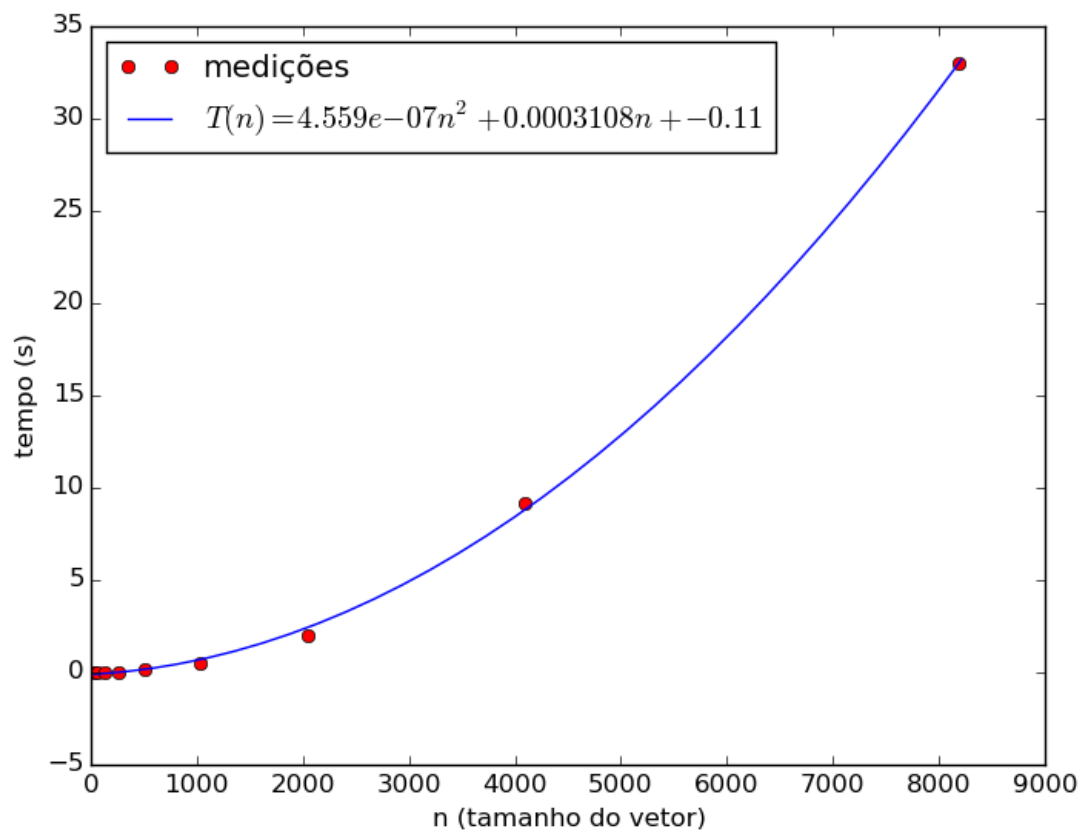


Figura 2.9: A análise do grafico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.559e - 07 * n^2 + 0.0003108 * n - 0.11$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.19568 * 10^{301}$

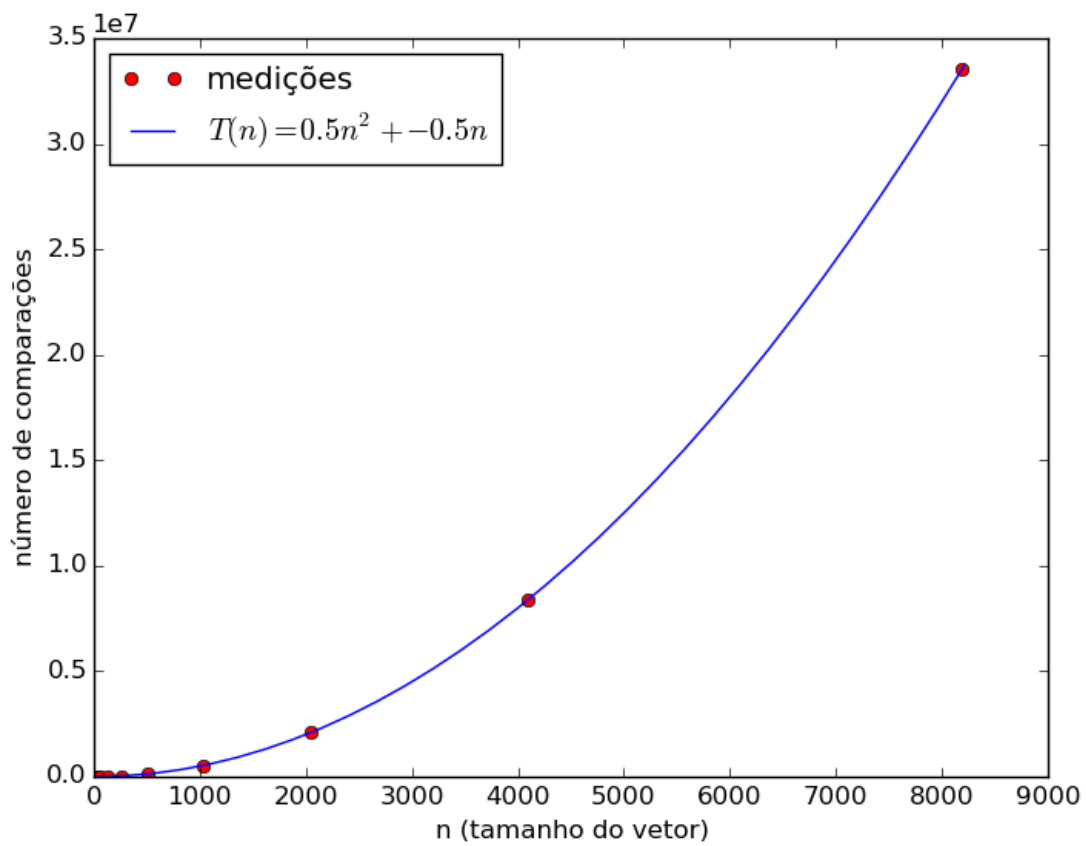


Figura 2.10: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000576
64	2016	0.002319
128	8128	0.008653
256	32640	0.033483
512	130816	0.131007
1024	523776	0.541596
2048	2096128	2.069160
4096	8386560	8.934160
8192	33550336	34.025600

Tabela 2.6: Tabela com vetor teste quase crescente 30%: A linha de interesse analisada para este caso é a 15.

n	comparações	tempo(s)
32	496	0.000602
64	2016	0.002394
128	8128	0.008284
256	32640	0.034522
512	130816	0.128441
1024	523776	0.520919
2048	2096128	2.154260
4096	8386560	8.764070
8192	33550336	34.113300

Tabela 2.7: Tabela com vetor teste quase crescente 40%: A linha de interesse analisada para este caso é a 15.

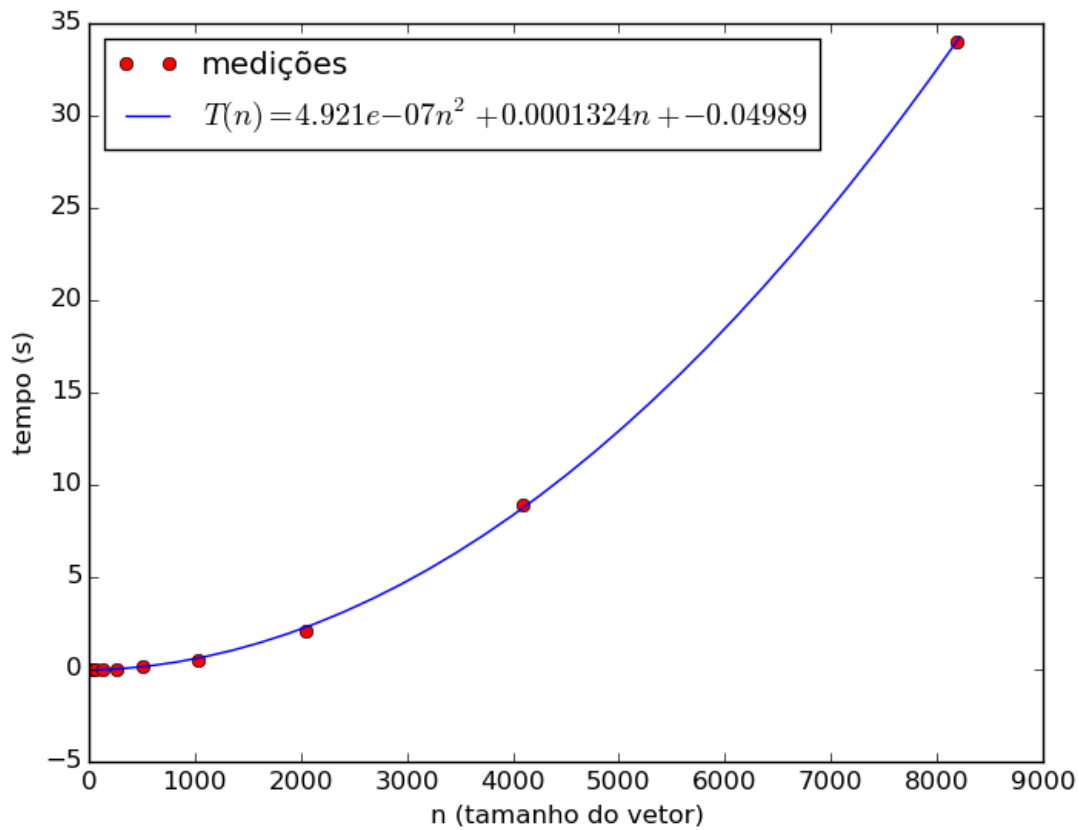


Figura 2.11: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.921e - 07 * n^2 + 0.0001324 * n - 0.04989$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.84645 * 10^{301}$

n	comparações	tempo(s)
32	496	0.000622
64	2016	0.002407
128	8128	0.008500
256	32640	0.036190
512	130816	0.140001
1024	523776	0.573178
2048	2096128	2.058500
4096	8386560	9.339260
8192	33550336	31.402100

Tabela 2.8: Tabela com vetor teste quase crescente 50%: A linha de interesse analisada para este caso é a 15.

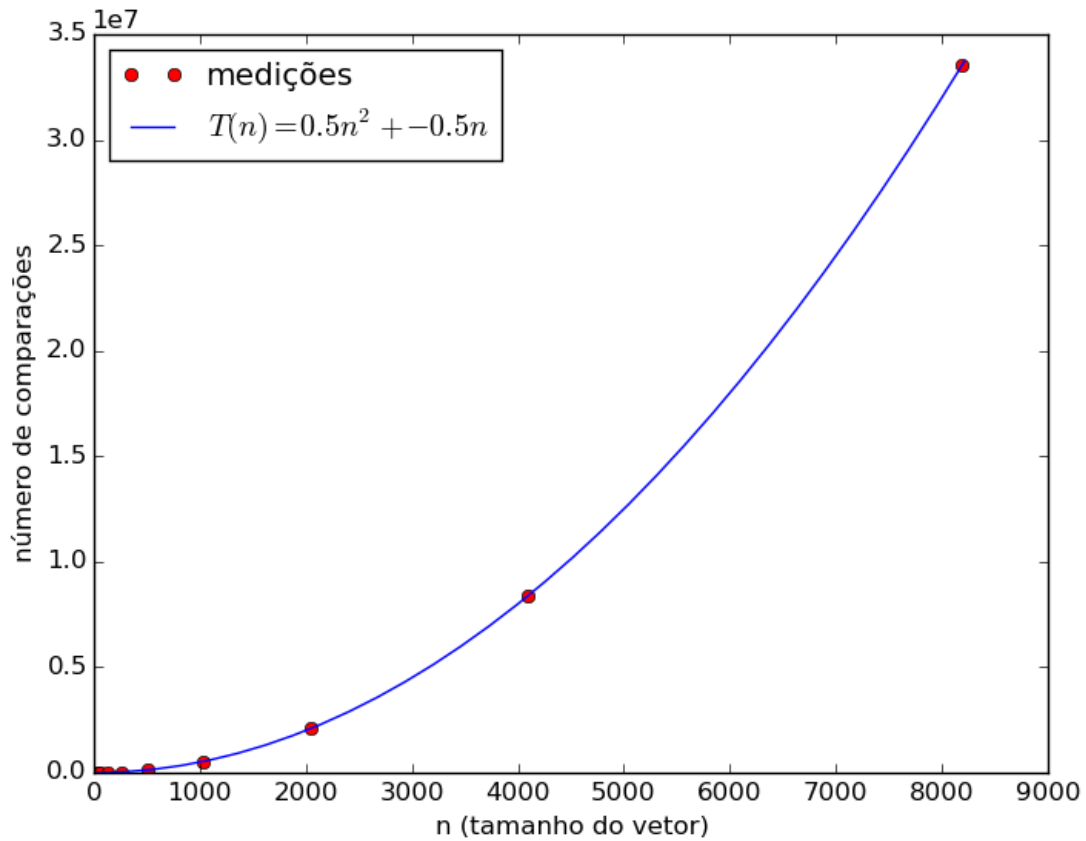


Figura 2.12: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000708
64	2016	0.002590
128	8128	0.009378
256	32640	0.038826
512	130816	0.155937
1024	523776	0.615690
2048	2096128	2.513880
4096	8386560	9.210180
8192	33550336	34.747800

Tabela 2.9: Tabela com vetor teste quase decrescente 10%: A linha de interesse analisada para este caso é a 15.

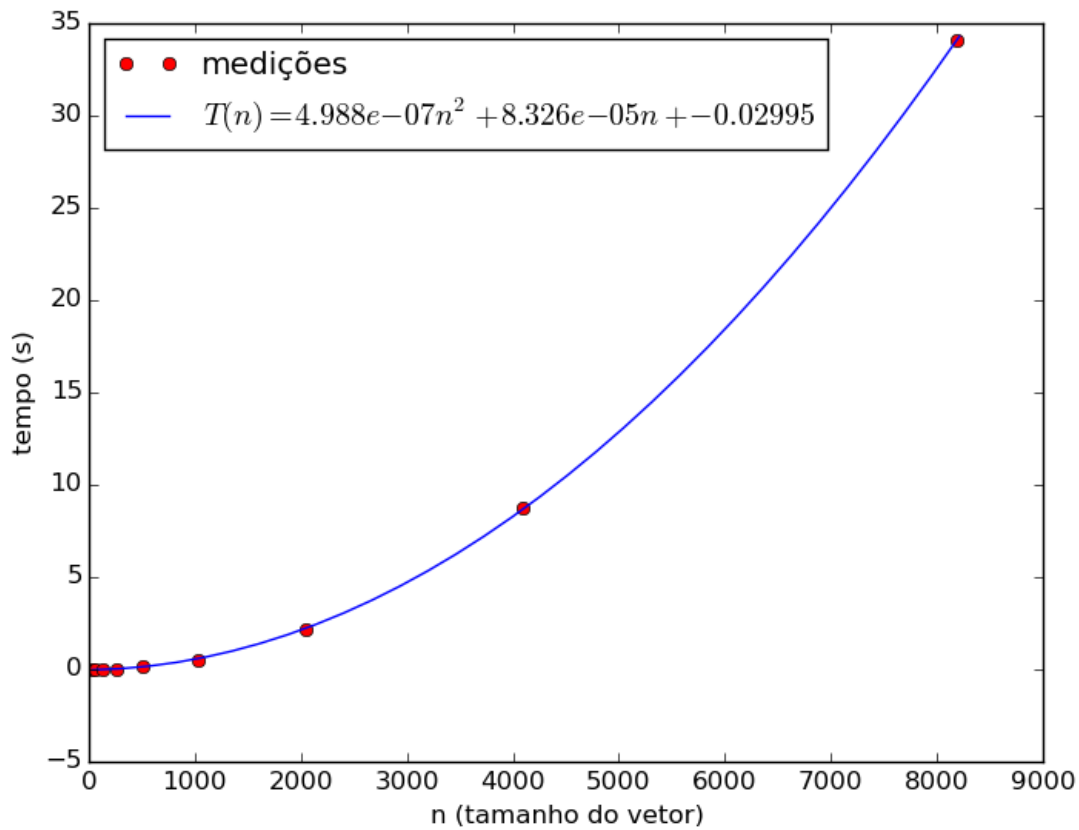


Figura 2.13: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.988e - 07 * n^2 + 8.326e - 5 * n - 0.02995$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.96689 * 10^{301}$

n	comparações	tempo(s)
32	496	0.000719
64	2016	0.002510
128	8128	0.008998
256	32640	0.038053
512	130816	0.136999
1024	523776	0.613878
2048	2096128	2.546130
4096	8386560	9.400370
8192	33550336	33.744400

Tabela 2.10: Tabela com vetor teste quase decrescente 20%: A linha de interesse analisada para este caso é a 15

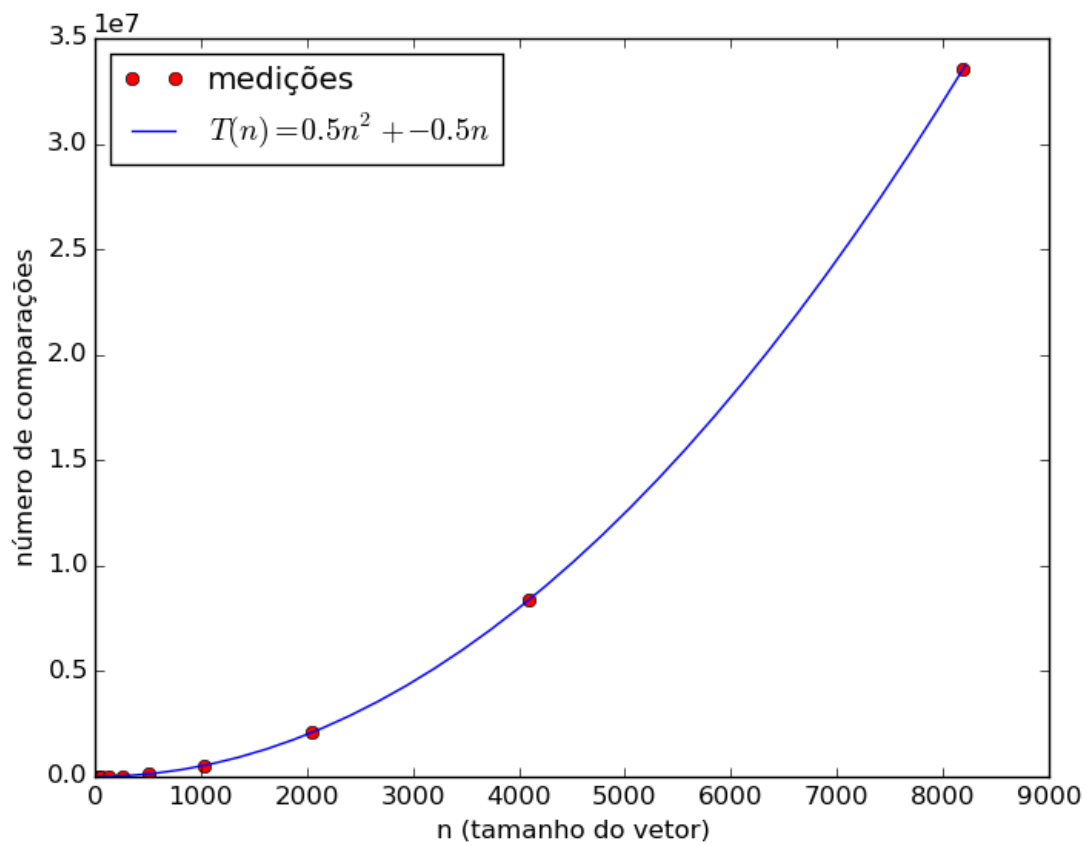


Figura 2.14: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

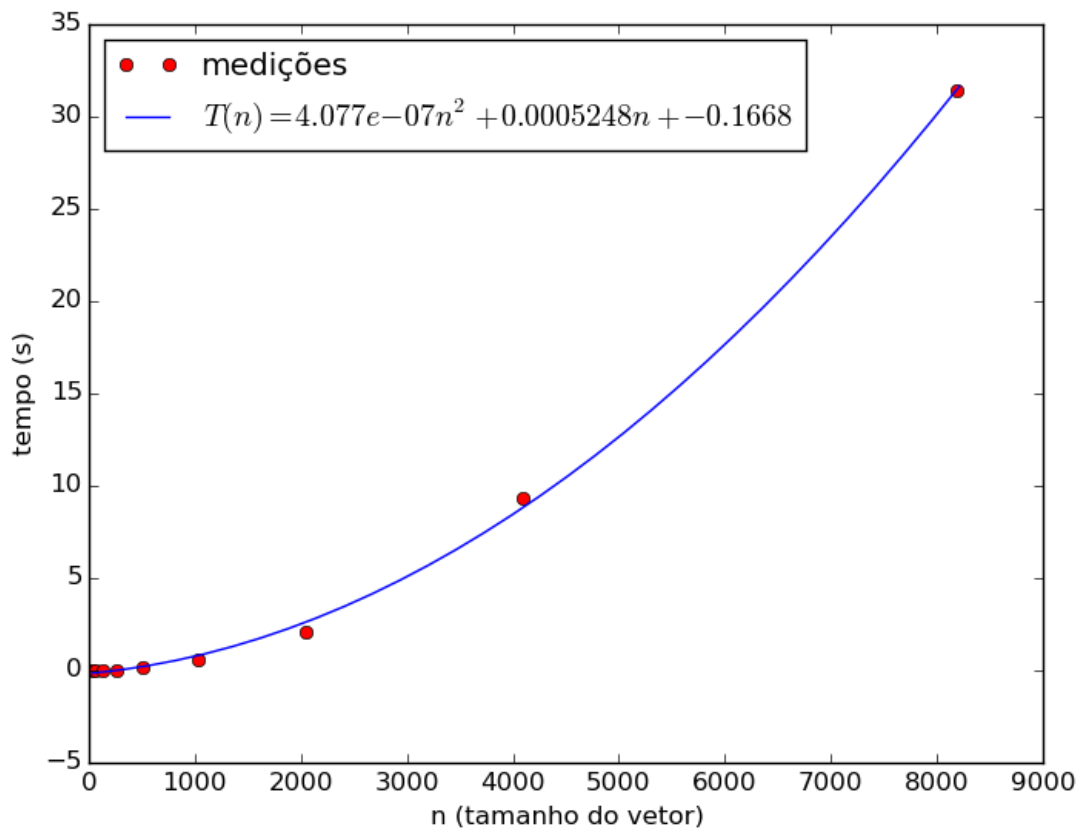


Figura 2.15: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.077e - 07 * n^2 + 0.0005248 * n - 0.1668$ e para o $n = 2^{32}$,
 $T(2^{32}) = 7.32919 * 10^{301}$

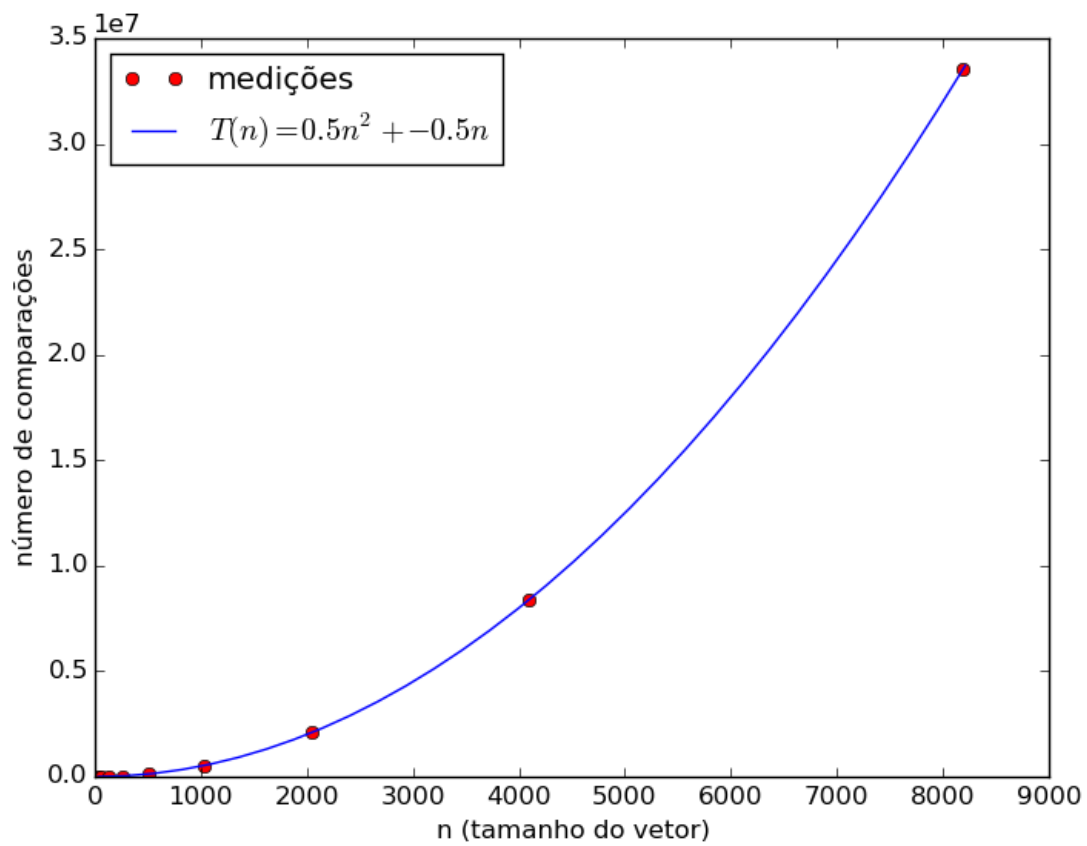


Figura 2.16: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

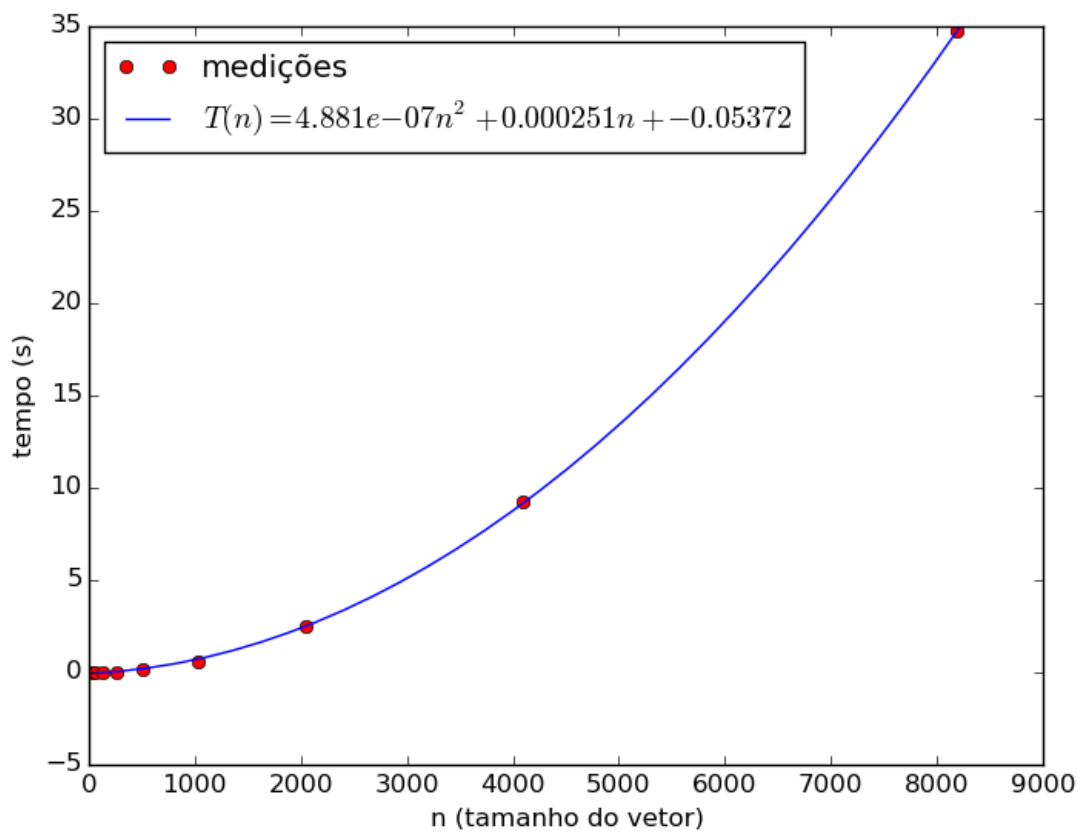
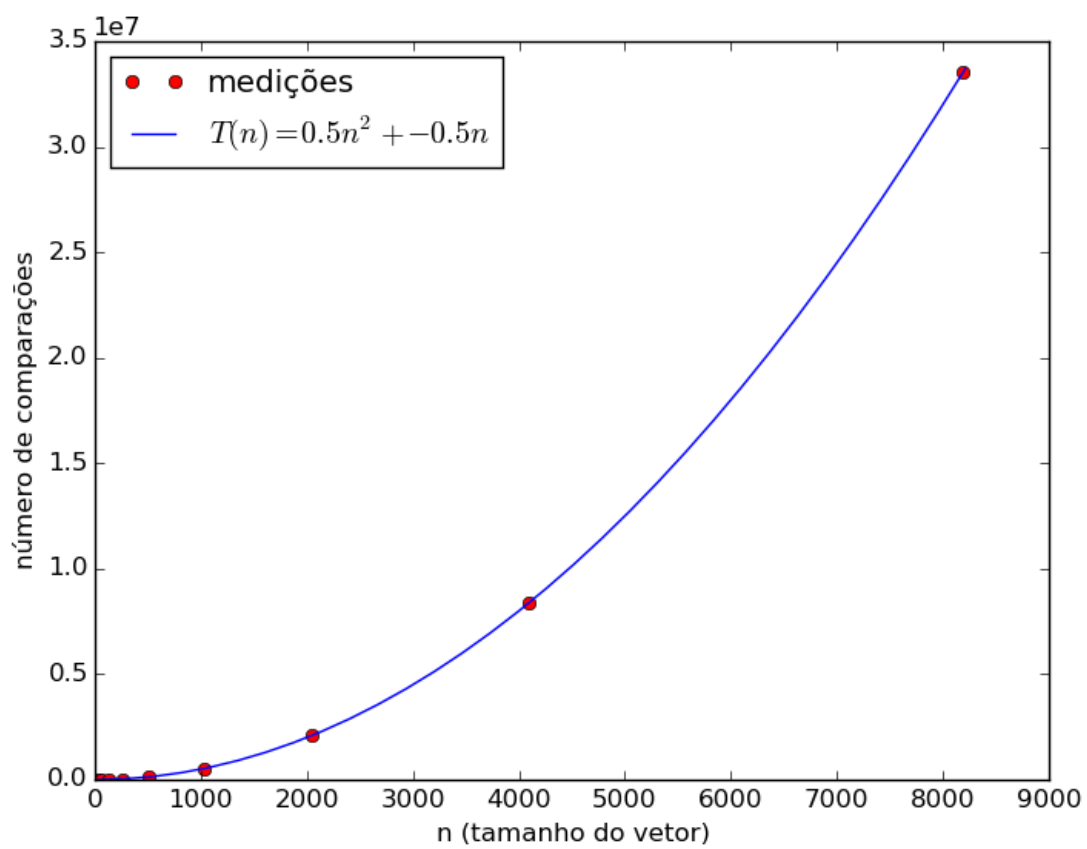


Figura 2.17: A análise do grafico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.881e - 07 * n^2 + 0.000251 * n - 0.05372$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.77454 * 10^{301}$



captionA análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

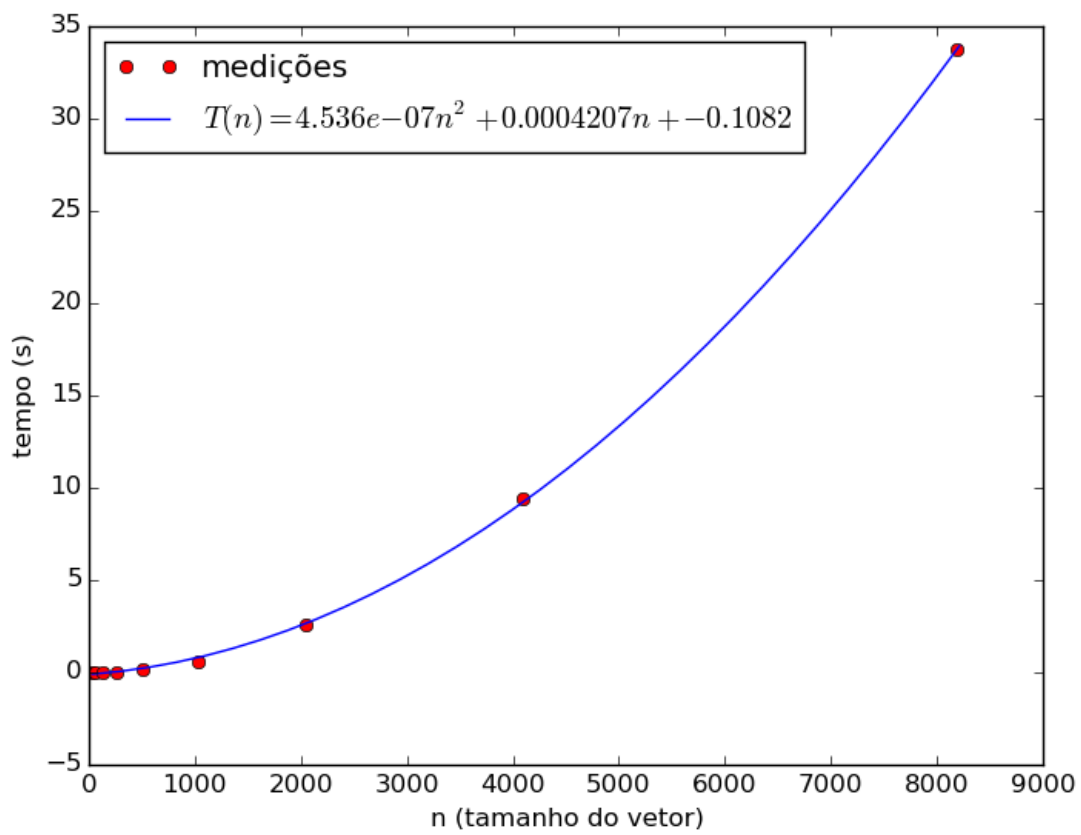


Figura 2.18: A análise do grafico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 4.536e - 07 * n^2 + 0.0004207 * n - 0.1082$ e para o $n = 2^{32}$,
 $T(2^{32}) = 8.15434 * 10^{301}$

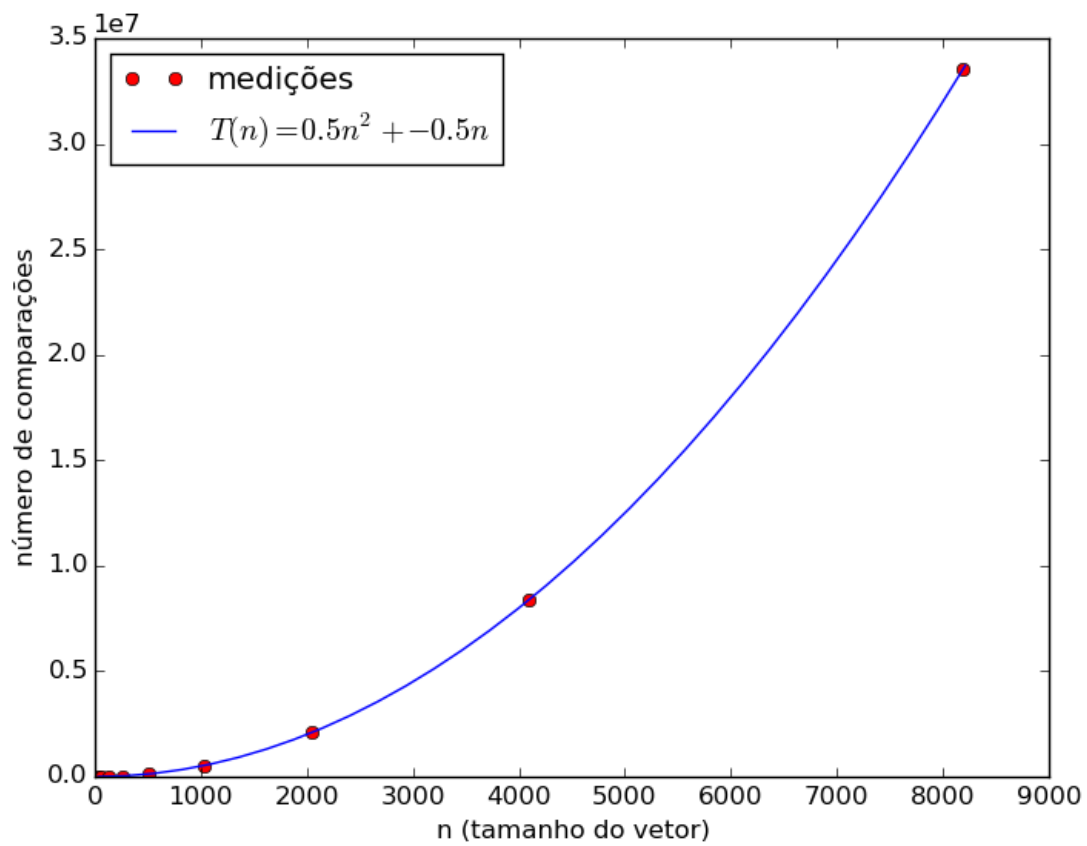


Figura 2.19: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000684
64	2016	0.002557
128	8128	0.009644
256	32640	0.038745
512	130816	0.144655
1024	523776	0.567072
2048	2096128	2.242760
4096	8386560	9.866570
8192	33550336	34.645600

Tabela 2.11: Tabela com vetor teste quase decrescente 30%: A linha de interesse analisada para este caso é a 15

n	comparações	tempo(s)
32	496	0.000655
64	2016	0.002451
128	8128	0.009061
256	32640	0.036720
512	130816	0.140110
1024	523776	0.625800
2048	2096128	2.254600
4096	8386560	8.999930
8192	33550336	35.220200

Tabela 2.12: Tabela com vetor teste quase decrescente 40%: A linha de interesse analisada para este caso é a 15

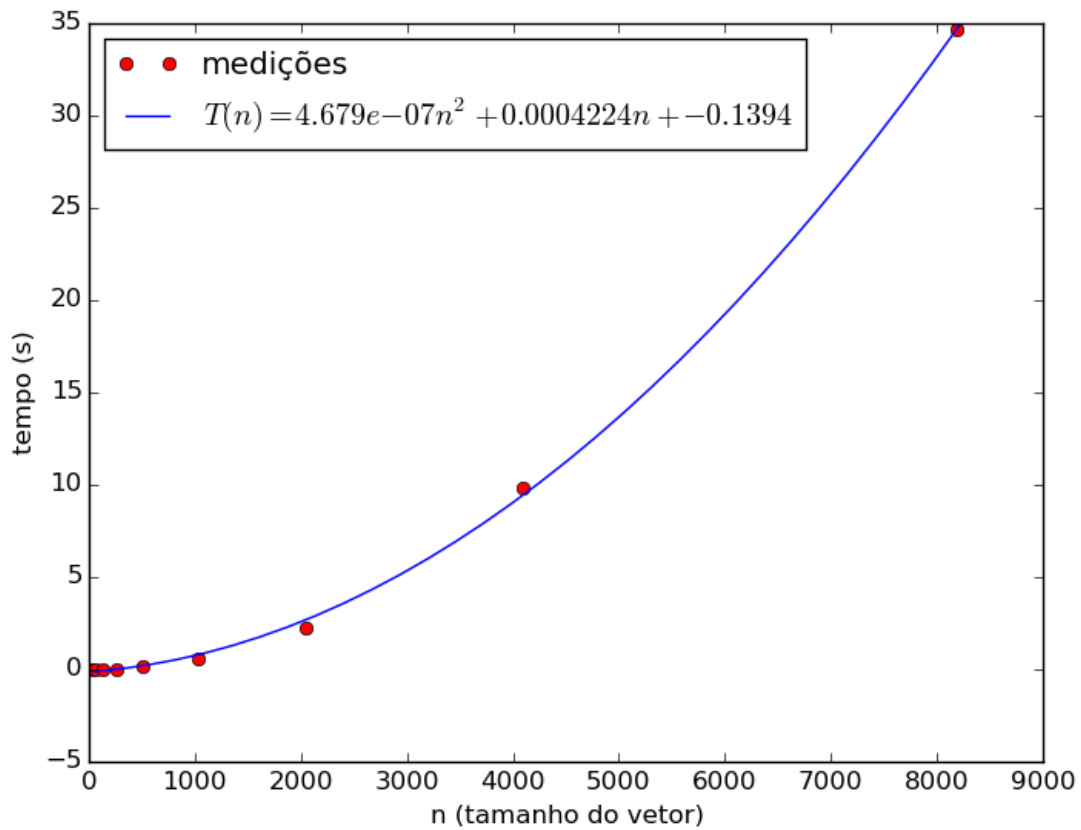


Figura 2.20: A análise do gráfico para 2^{32} segue abaixo para selectionsort. Tendo a função $T(n) = 4.679 * n^2 - 07 * n - 0.1394$ e para o $n = 2^{32}$, $T(2^{32}) = 8.411406178020777 * 10^{308}$

n	comparações	tempo(s)
32	496	0.000679
64	2016	0.002268
128	8128	0.009045
256	32640	0.034093
512	130816	0.141664
1024	523776	0.531386
2048	2096128	2.157350
4096	8386560	8.908460
8192	33550336	35.002700

Tabela 2.13: Tabela com vetor teste quase decrescente 50%: A linha de interesse analisada para este caso é a 15

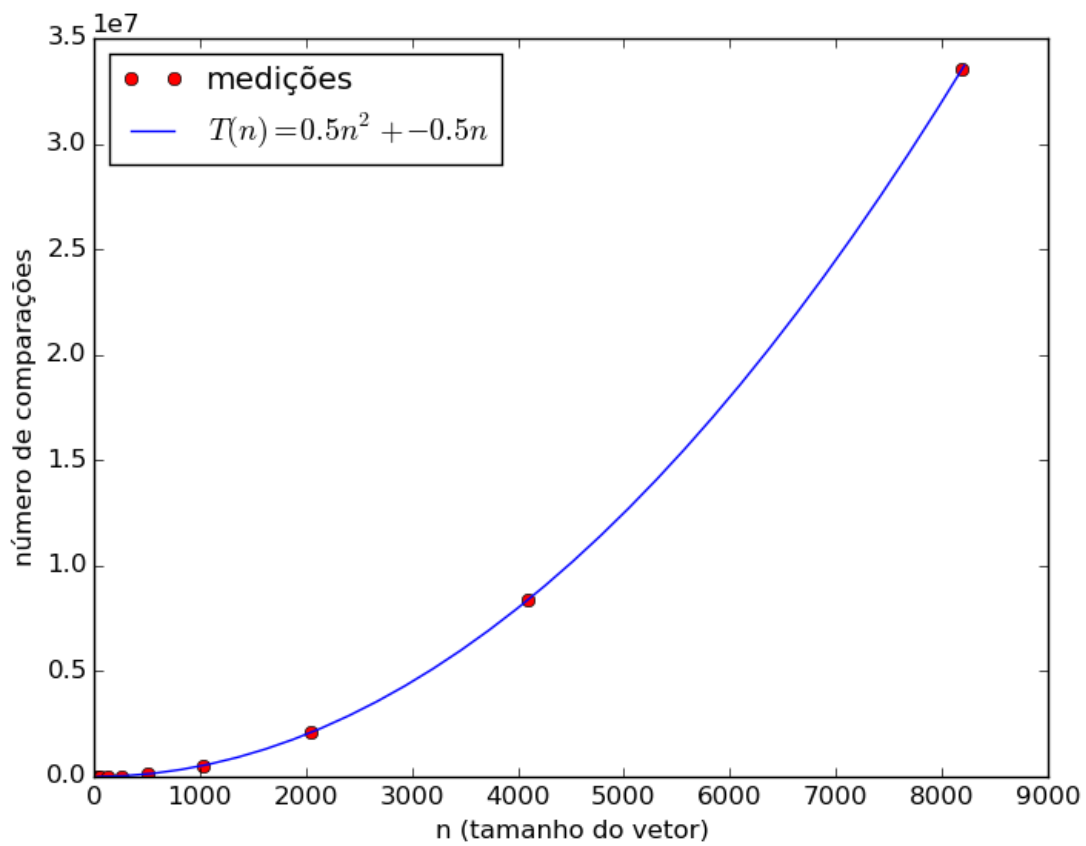


Figura 2.21: A análise do grafico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

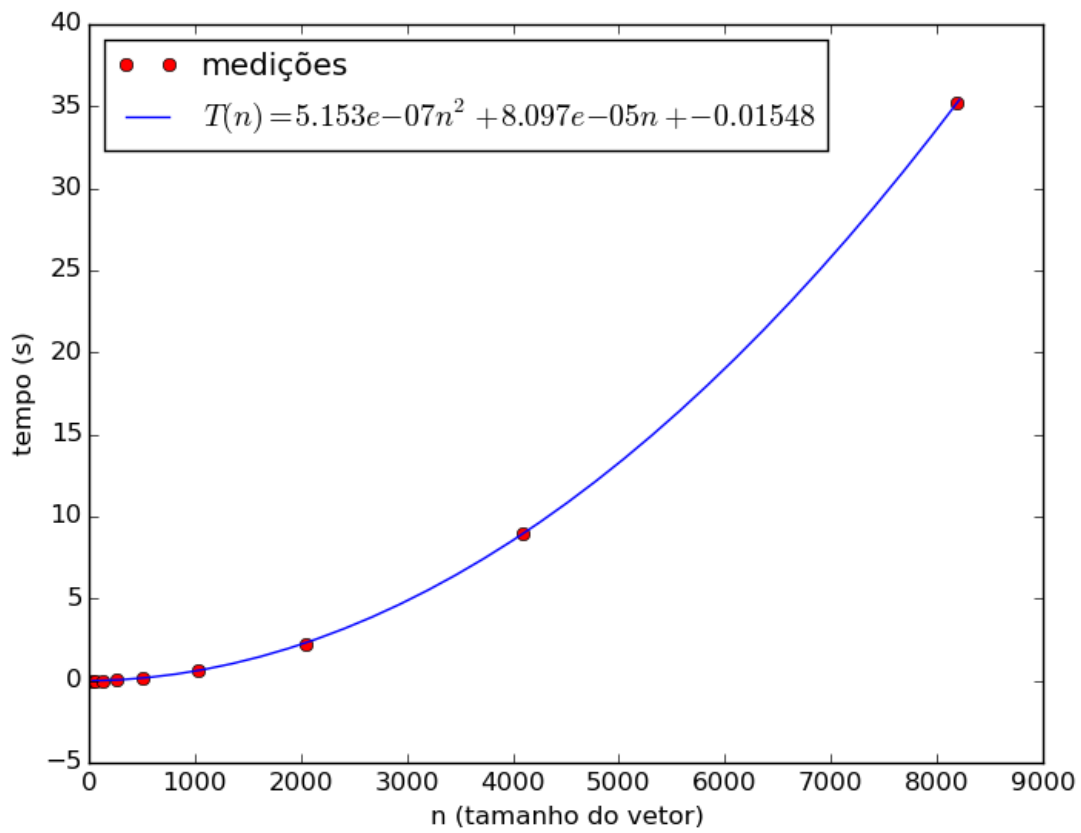


Figura 2.22: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 5.153e-07 * n^2 + 8.097e-5 * n - 0.01548$ e para o $n = 2^{32}$,
 $T(2^{32}) = 9.26351272 * 10^{301}$

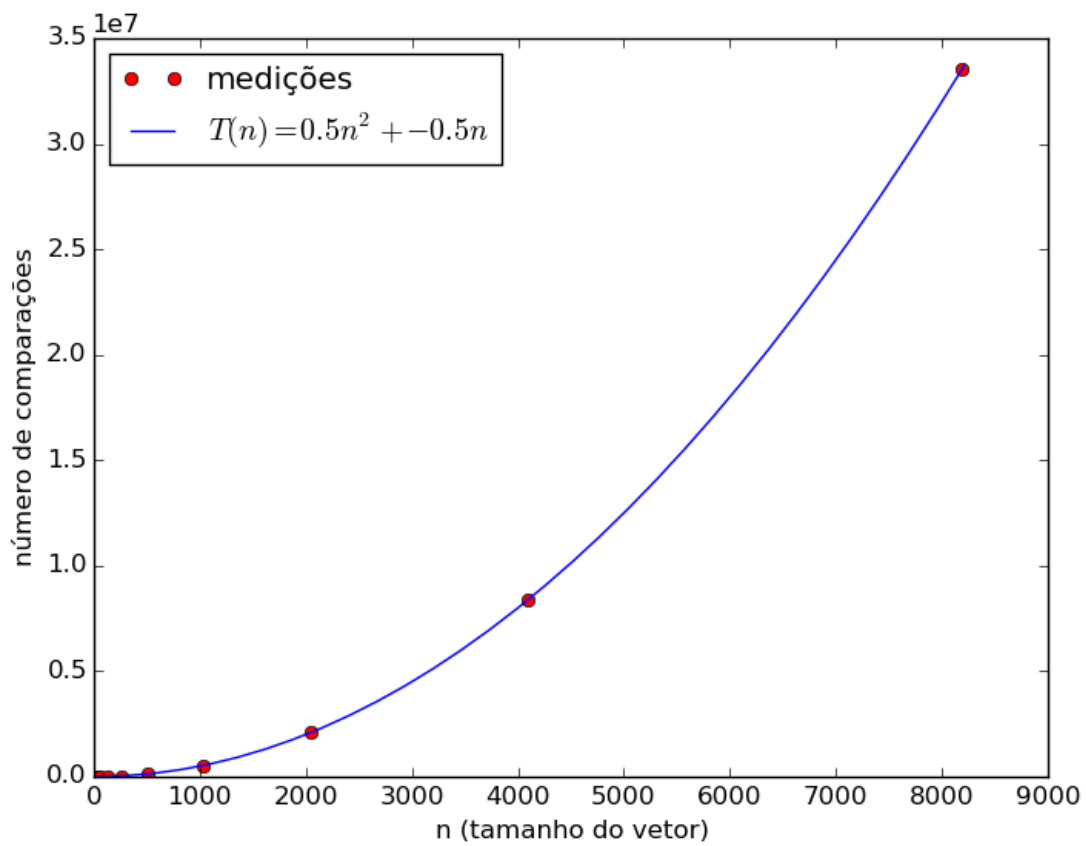


Figura 2.23: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

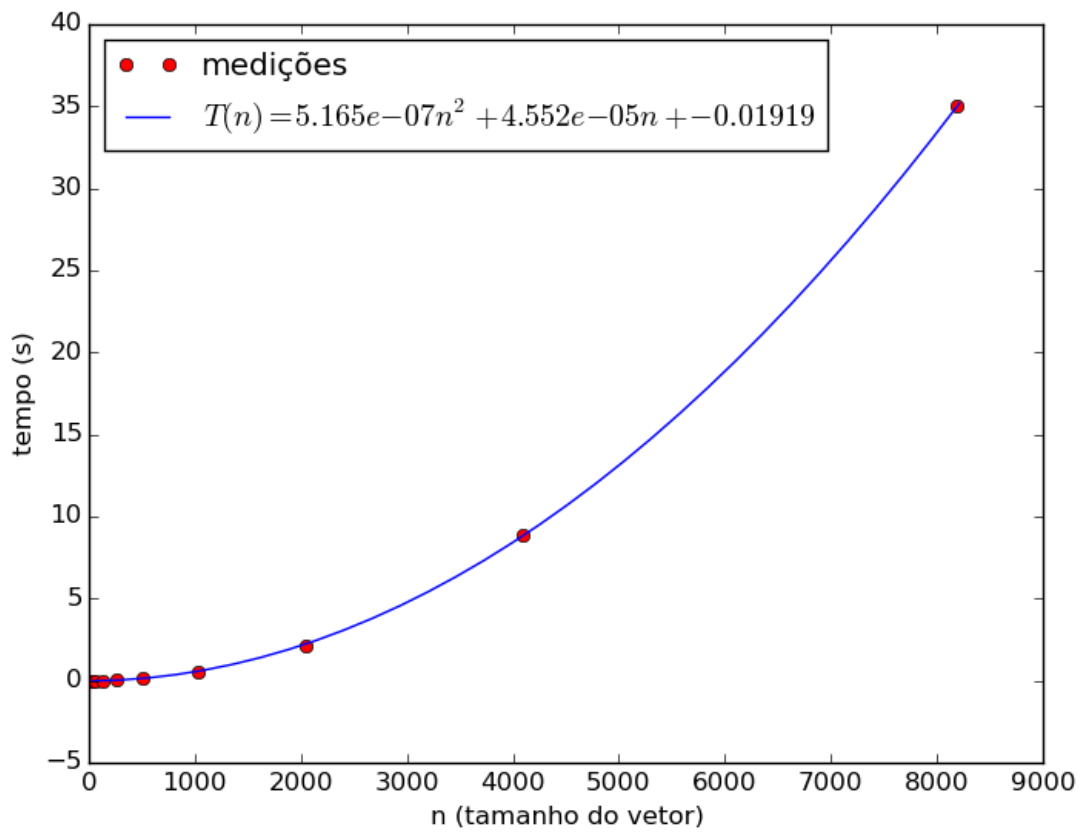


Figura 2.24: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 5.165e-07 * n^2 + 4.552e-5 * n - 0.01919$ e para o $n = 2^{32}$,
 $T(2^{32}) = 9.2850850415 * 10^{301}$

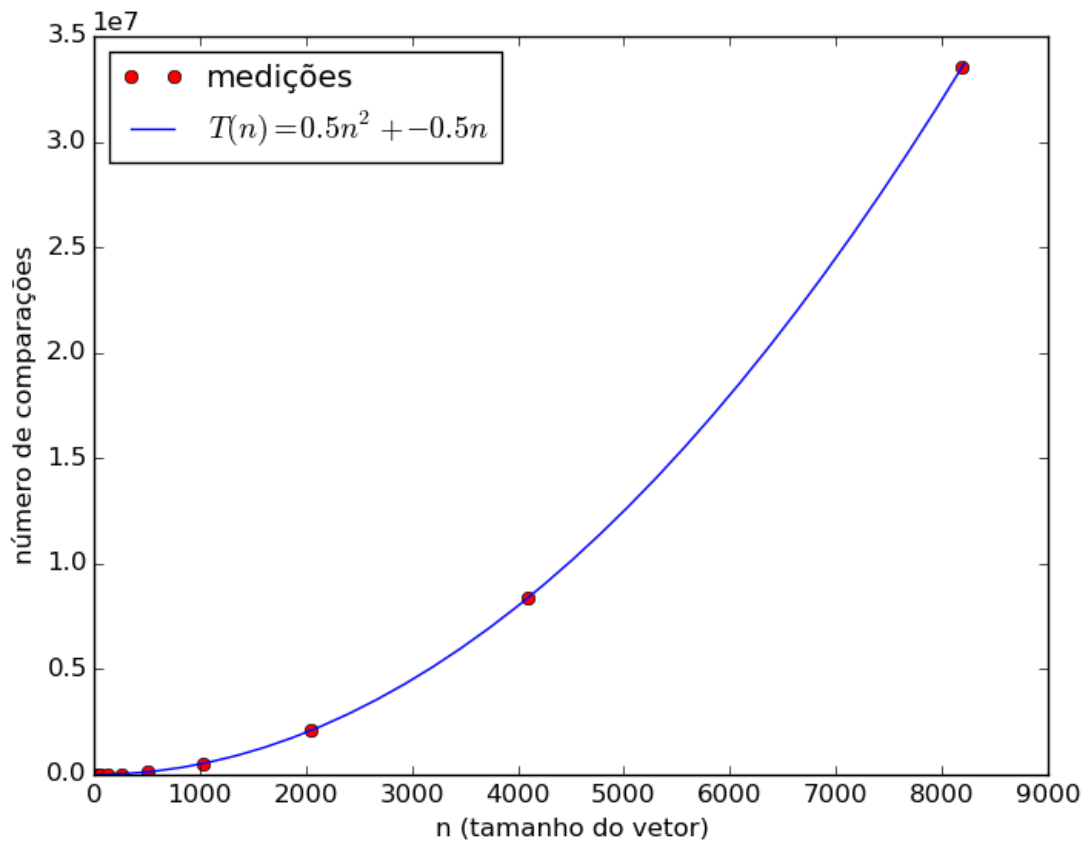


Figura 2.25: A análise do gráfico para 2^{32} segue abaixo para selectionsort
Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884656 * 10^{307}$

Apêndice A

Arquivo ../selectionsort/selectionsort.py

Listagem A.1: ../selectionsort/selectionsort.py

```
1 import numpy as np
2
3 @profile
4 def selectionsort( lista ):
5     for i in range( len( lista ) ):
6         least = i
7         for k in range( i + 1 , len( lista ) ):
8             if lista[k] < lista[least]:
9                 least = k
10        troca( lista, least, i )
11
12 def troca( A, x, y ):
13     tmp = A[x]
14     A[x] = A[y]
15     A[y] = tmp
```

Apêndice B

Arquivo ../selectionsort/ensaio.py

Listagem B.1: ../selectionsort/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)
```
