

Análise experimental de algoritmos usando Python

Patricia 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 bubblesort de vetor aleatório. Tendo a função $T(n) = 9.314e - 07 * n^2 + 0.0001364 * n - 0.02926$ e para o $n = 2^{32}$, $T(2^{32}) = 1.6743713 * 10^{302}$	9
2.2	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor aleatório. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	10
2.3	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor crescente. Tendo a função $T(n) = 5.427e - 07 * n^2 + 3.706e - 05 * n - 0.02506$ e para o $n = 2^{32}$, $T(2^{32}) = 9.75608 * 10^{301}$	11
2.4	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor crescente. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	12
2.5	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor decrescente. Tendo a função $T(n) = 9.314e - 07 * n^2 + 0.0001364 * n - 0.02926$ e para o $n = 2^{32}$, $T(2^{32}) = 1.674 * 10^{302}$	13
2.6	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor decrescente. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	14
2.7	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 10%. Tendo a função $T(n) = 1.226e - 06 * n^2 + 0.0004003 * n - 0.1198$ e para o $n = 2^{32}$, $T(2^{32}) = 2.203 * 10^{302}$	15
2.8	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 10%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	16
2.9	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 20%. Tendo a função $T(n) = 1.639e - 06 * n^2 + 0.001244 * n - 0.3636$ e para o $n = 2^{32}$, $T(2^{32}) = 2.94641 * 10^{302}$	17
2.10	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 20%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	18
2.11	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 30%. Tendo a função $T(n) = 1.305e - 06 * n^2 + 0.0002128 * n - 0.08068$ e para o $n = 2^{32}$, $T(2^{32}) = 2.345989 * 10^{302}$	20
2.12	A análise em número de comparações do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 30%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	21
2.13	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 40%. Tendo a função $T(n) = 1.425e - 06 * n^2 + 0.0004043 * n - 0.09253$ e para o $n = 2^{32}$, $T(2^{32}) = 2.56171 * 10^{302}$	22

2.14	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase crescente 40%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	23
2.15	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 50%. Tendo a função $T(n) = 1.211e-06 * n^2 + 0.0003953 * n - 0.1445$ e para o $n = 2^{32}$, $T(2^{32}) = 2.177006 * 10^{302}$	24
2.16	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase crescente 50%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	25
2.17	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 10%. Tendo a função $T(n) = 5.404e-07 * n^2 - e-05 * n - 0.03211$ e para o $n = 2^{32}$, $T(2^{32}) = 9.71473 * 10^{301}$	26
2.18	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase decrescente 10%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	27
2.19	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 20%. Tendo a função $T(n) = 5.404e-07 * n^2 - e-05 * n - 0.0632$ e para o $n = 2^{32}$, $T(2^{32}) = 1.116906 * 10^{302}$	28
2.20	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase decrescente 20%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	29
2.21	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 30%. Tendo a função $T(n) = 5.637e-07 * n^2 - 1.65e-5 * n - 0.01043$ e para o $n = 2^{32}$, $T(2^{32}) = 1.0133596 * 10^{302}$	31
2.22	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase decrescente 30%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	32
2.23	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 40%. Tendo a função $T(n) = 6.457e-07 * n^2 - 0.00019495 * n - 0.05777$ e para o $n = 2^{32}$, $T(2^{32}) = 1.1607704 * 10^{302}$	33
2.24	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase decrescente 40%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	34
2.25	A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 50%. Tendo a função $T(n) = 5.483e-07 * n^2 + 0.0002625 * n - 0.06923$ e para o $n = 2^{32}$, $T(2^{32}) = 9.856751 * 10^{301}$	35
2.26	A análise em número de comparações do grafico para 2^{32} segue abaixo para bobblesort de vetor quase decrescente 50%. Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$	36

Lista de Tabelas

2.1	Tabela com vetor teste aleatório: a linha de interesse analisada para este caso é a 15	8
2.2	Tabela com vetor teste crescente: a linha de interesse analisada para este caso é a 15	8
2.3	Tabela com vetor teste decrescente: a linha de interesse analisada para este caso é a 15	9
2.4	Tabela com vetor teste quase crescente 10%: a linha de interesse analisada para este caso é a 15	10
2.5	Tabela com vetor teste quase crescente 20%: a linha de interesse analisada para este caso é a 15	11
2.6	Tabela com vetor teste quase crescente 30%: a linha de interesse analisada para este caso é a 15	19
2.7	Tabela com vetor teste quase crescente 40%: a linha de interesse analisada para este caso é a 15	19
2.8	Tabela com vetor teste quase crescente 50%: a linha de interesse analisada para este caso é a 15	20
2.9	Tabela com vetor teste quase decrescente 10%: a linha de interesse analisada para este caso é a 15	21
2.10	Tabela com vetor teste quase decrescente 20%: a linha de interesse analisada para este caso é a 15	22
2.11	Tabela com vetor teste quase decrescente 30%: a linha de interesse analisada para este caso é a 15	30
2.12	Tabela com vetor teste quase decrescente 40%: a linha de interesse analisada para este caso é a 15	30
2.13	Tabela com vetor teste quase decrescente 50%: a linha de interesse analisada para este caso é a 15	31

Lista de Listagens

A.1	../bolha/bolha.py	37
B.1	../bolha/ensaio.py	38

Sumário

Lista de Figuras	2
Lista de Tabelas	4
1 Análise	7
1.0.1 Introdução	7
1.0.2 Desempenho do bubblesort	7
2 Resultados	8
2.1 Tabelas	8
Apêndice	37
A Arquivo ../bolha/bolha.py	37
B Arquivo ../bolha/ensaio.py	38

Capítulo 1

Análise

1.0.1 Introdução

O bubblesort ou método da bolha, é um algoritmo de ordenação bastante simples, onde comparam-se dois elementos e trocam-se suas posições se o segundo elemento é menor do que o primeiro. Nesse algoritmo são feitas várias passagens pelo vetor comparando dois elementos adjacentes, se esses elementos estiverem fora de ordem, eles são trocados.

Vantagens:

Simplicidade do algoritmo, estável Desvantagens:

Lentidão

Indicações:

Vetores pequenos, demonstrações didáticas

1.0.2 Desempenho do bubblesort

As operações de comparação e de troca de n elementos são executadas no pior caso, o algoritmo realizará $n-1$ troca para o primeiro passo, e depois $n-2$ trocas para o segundo elemento e assim sucessivamente. Trocas = $n-1+n-2+n-2+\dots+2+1$ aproximadamente n^2 trocas. No melhor caso nenhuma troca será realizada, pois em ambos os casos o algoritmo faz da ordem, portanto:

Complexidade de espaço: o algoritmo tem complexidade de espaço referente ao tamanho do vetor de dados.

Complexidade de tempo: tem maior custo em comparações e troca de posições, assim a complexidade de tempo é

$$T(n) = O(n^2)$$

Capítulo 2

Resultados

2.1 Tabelas

n	comparações	tempo(s)
32	496	0.000966
64	2016	0.004182
128	8128	0.014954
256	32640	0.061451
512	130816	0.236011
1024	523776	1.053630
2048	2096128	4.185620
4096	8386560	16.159700
8192	33550336	63.593300

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	0.000565
64	2016	0.002440
128	8128	0.008909
256	32640	0.035941
512	130816	0.142118
1024	523776	0.581946
2048	2096128	2.101060
4096	8386560	9.395360
8192	33550336	36.670900

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

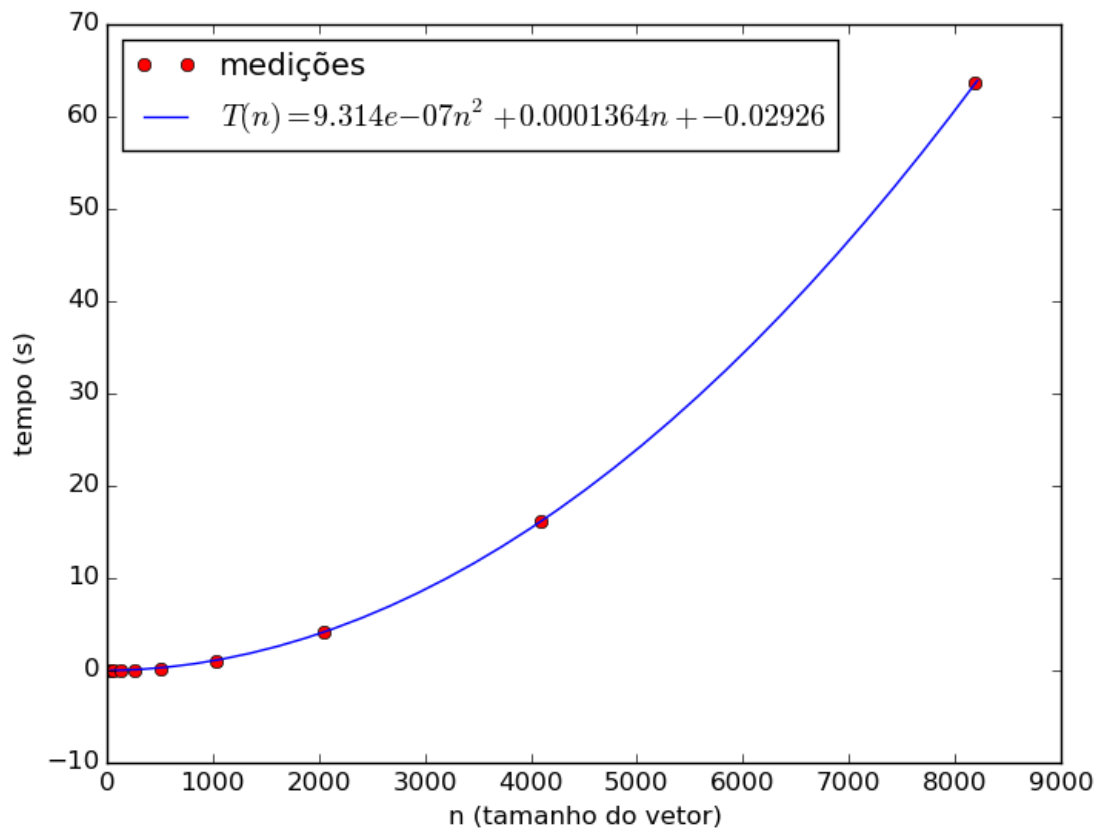


Figura 2.1: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor aleatório. Tendo a função $T(n) = 9.314e - 07 * n^2 + 0.0001364 * n - 0.02926$ e para o $n = 2^{32}$, $T(2^{32}) = 1.6743713 * 10^{302}$

n	comparações	tempo(s)
32	496	0.001386
64	2016	0.005775
128	8128	0.022430
256	32640	0.088820
512	130816	0.383981
1024	523776	1.439680
2048	2096128	5.957860
4096	8386560	21.605700
8192	33550336	93.107000

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

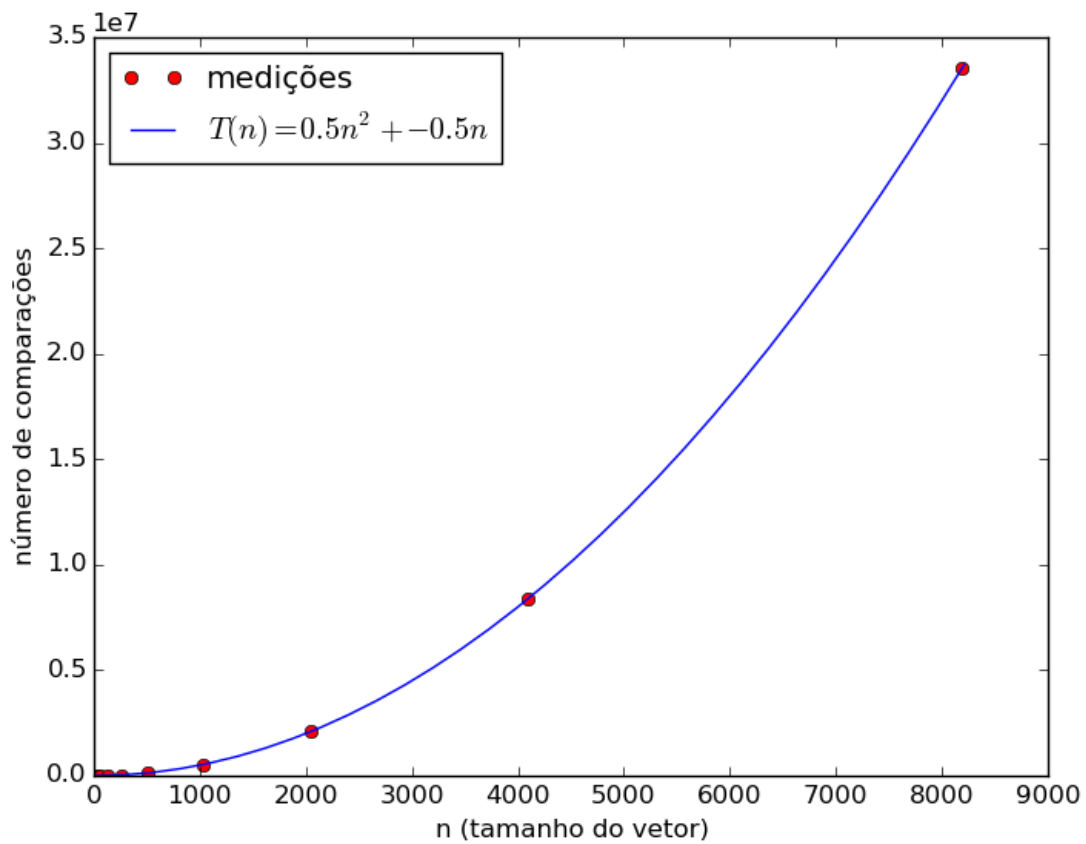


Figura 2.2: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bobblesort de vetor aleatório.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000574
64	2016	0.002152
128	8128	0.009420
256	32640	0.033514
512	130816	0.163590
1024	523776	0.554991
2048	2096128	2.162600
4096	8386560	8.656440
8192	33550336	35.559200

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

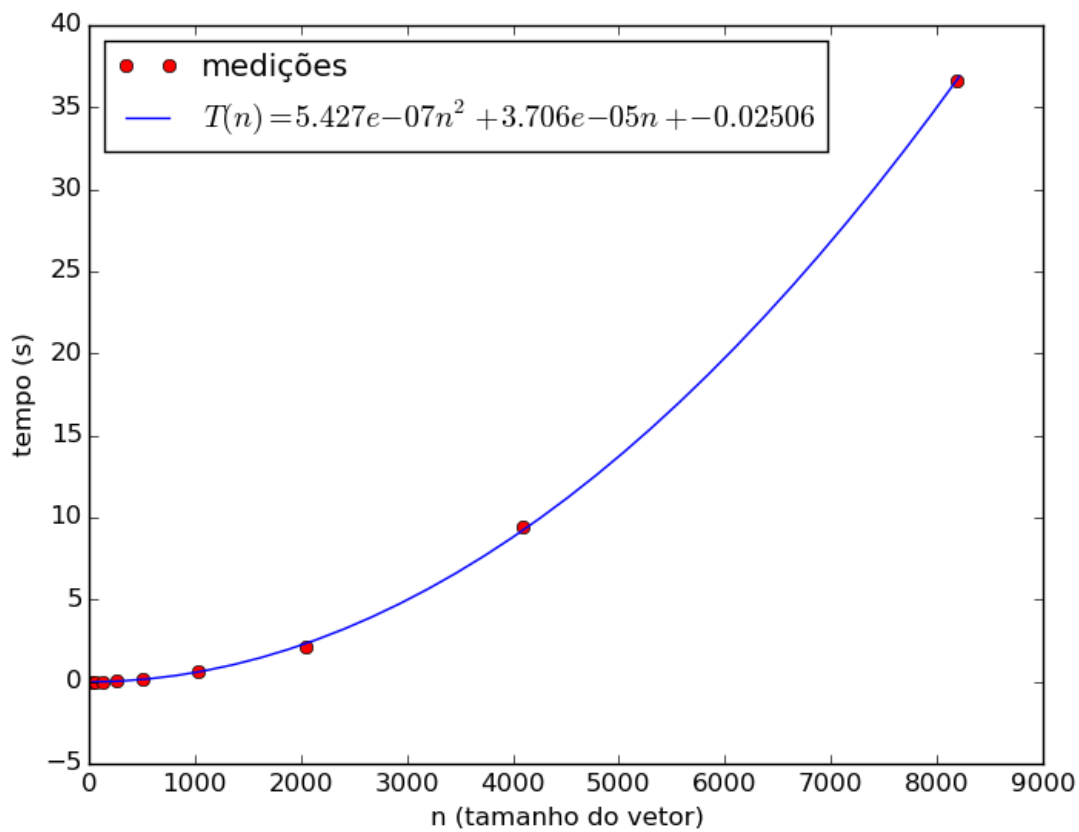


Figura 2.3: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor crescente. Tendo a função $T(n) = 5.427e - 07 * n^2 + 3.706e - 05 * n - 0.02506$ e para o $n = 2^{32}$, $T(2^{32}) = 9.75608 * 10^{301}$

n	comparações	tempo(s)
32	496	0.000580
64	2016	0.002290
128	8128	0.009233
256	32640	0.036300
512	130816	0.143595
1024	523776	0.555577
2048	2096128	2.317520
4096	8386560	9.365210
8192	33550336	39.829700

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

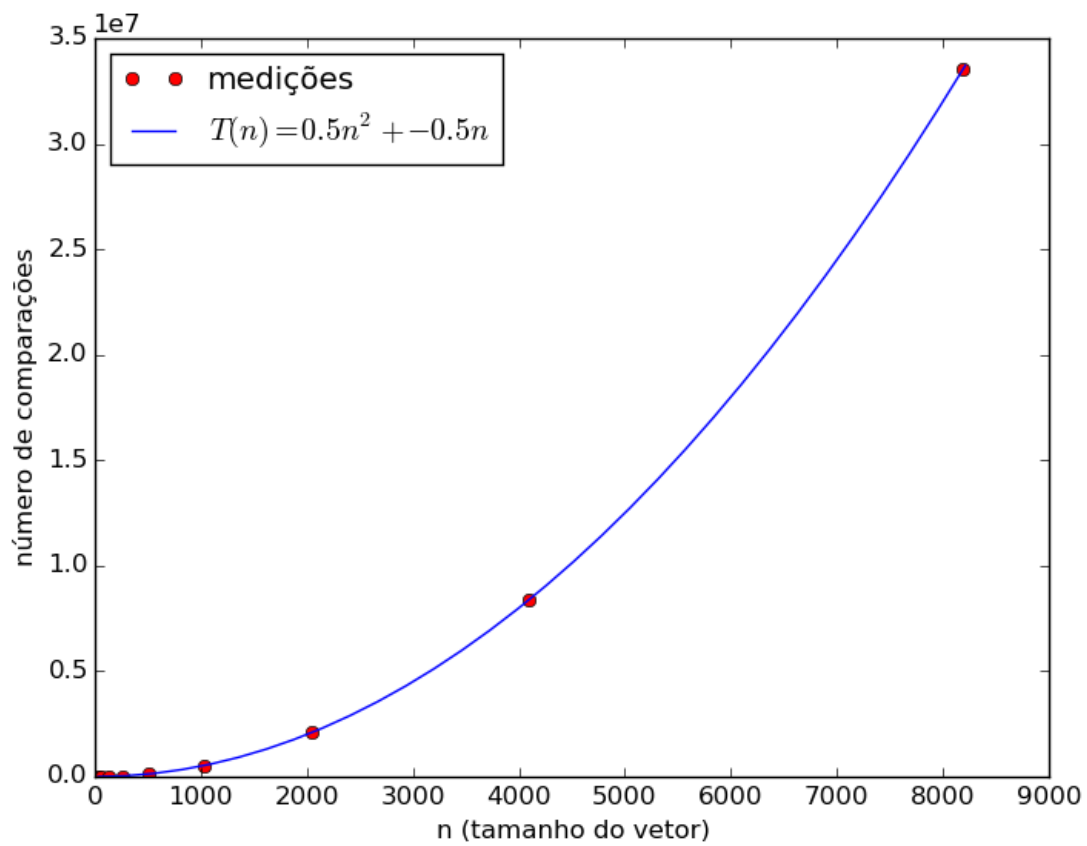


Figura 2.4: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor crescente.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

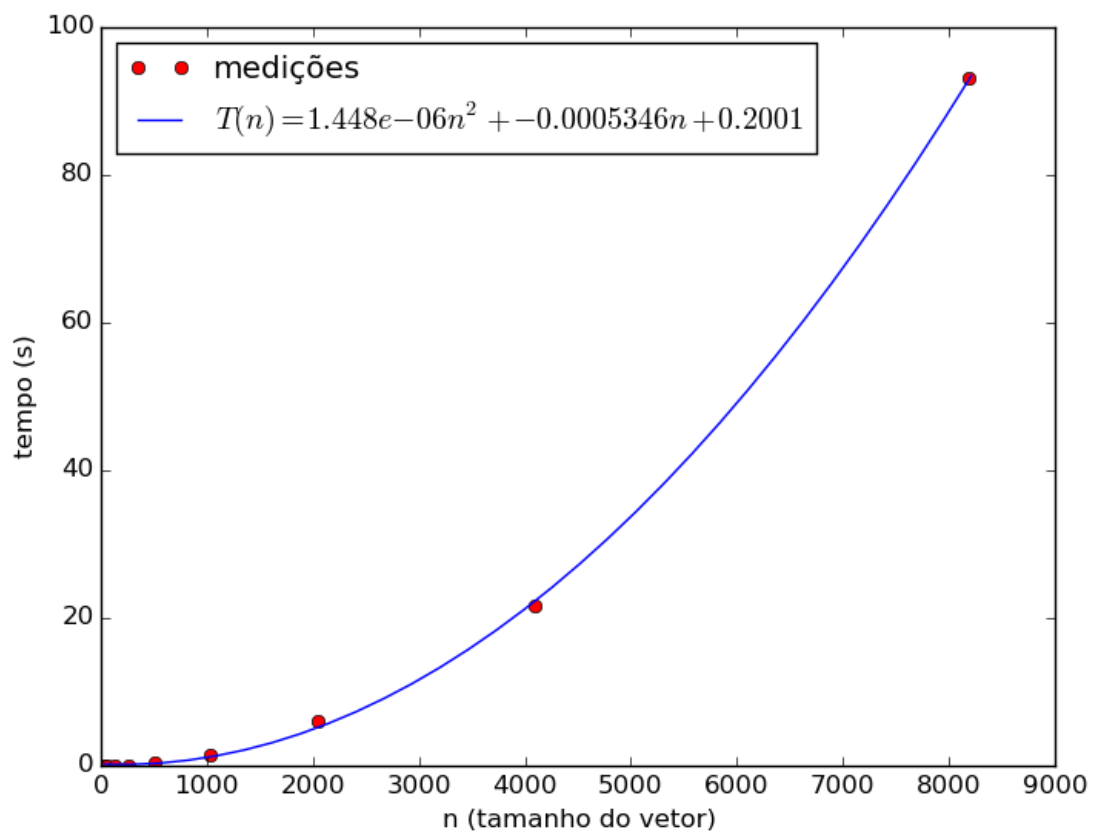


Figura 2.5: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor decrescente. Tendo a função $T(n) = 9.314e - 07 * n^2 + 0.0001364 * n - 0.02926$ e para o $n = 2^{32}$, $T(2^{32}) = 1.674 * 10^{302}$

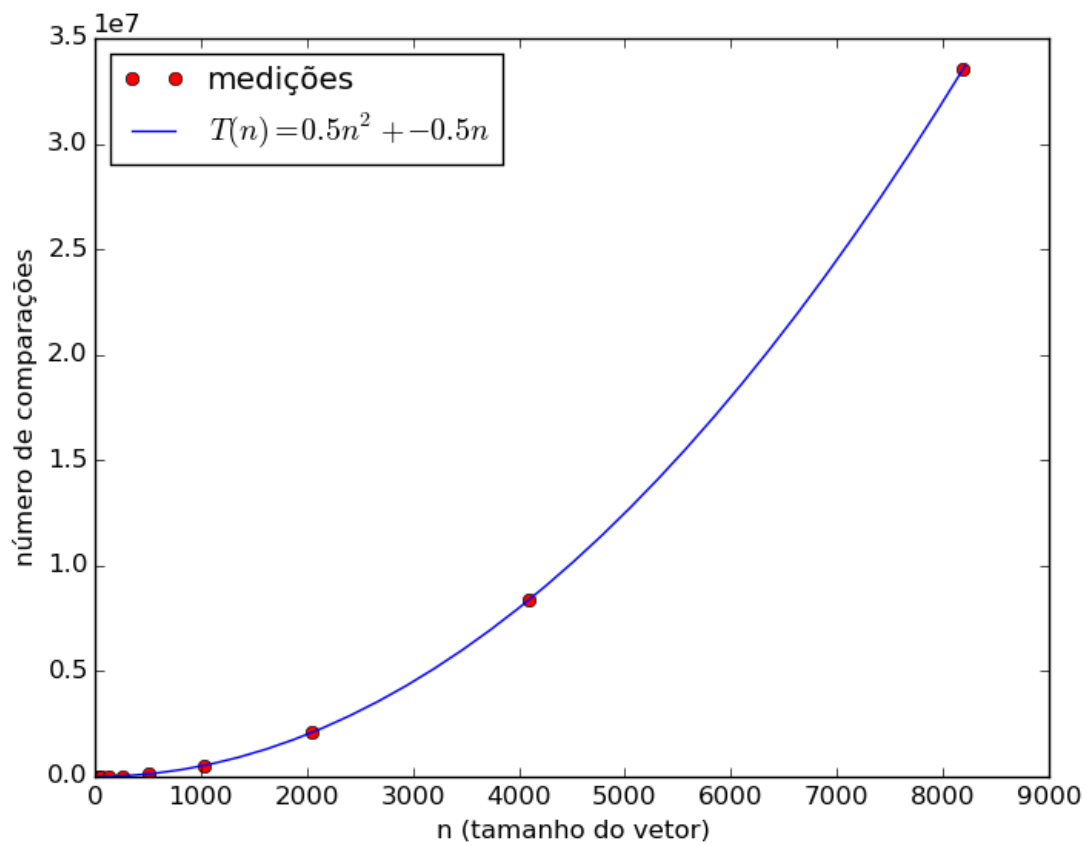


Figura 2.6: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor decrescente.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

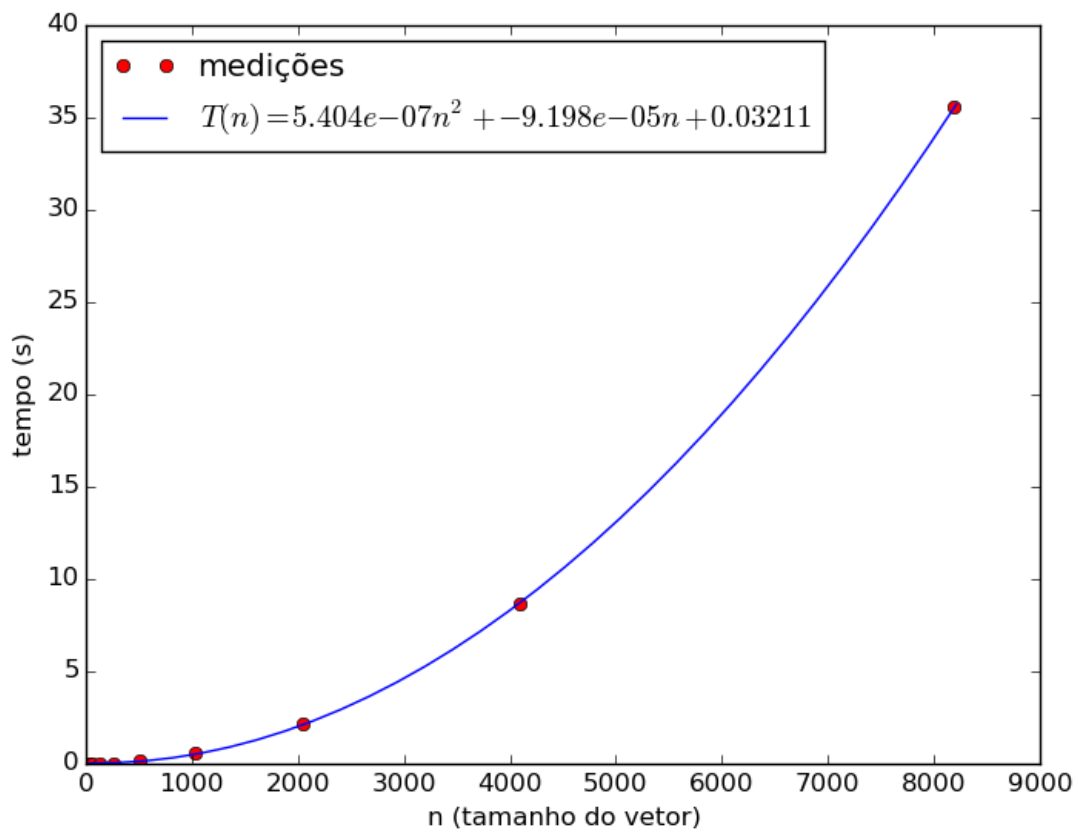


Figura 2.7: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 10%.

Tendo a função $T(n) = 1.226e-06*n^2 + 0.0004003*n - 0.1198$ e para o $n = 2^{32}$, $T(2^{32}) = 2.203*10^{302}$

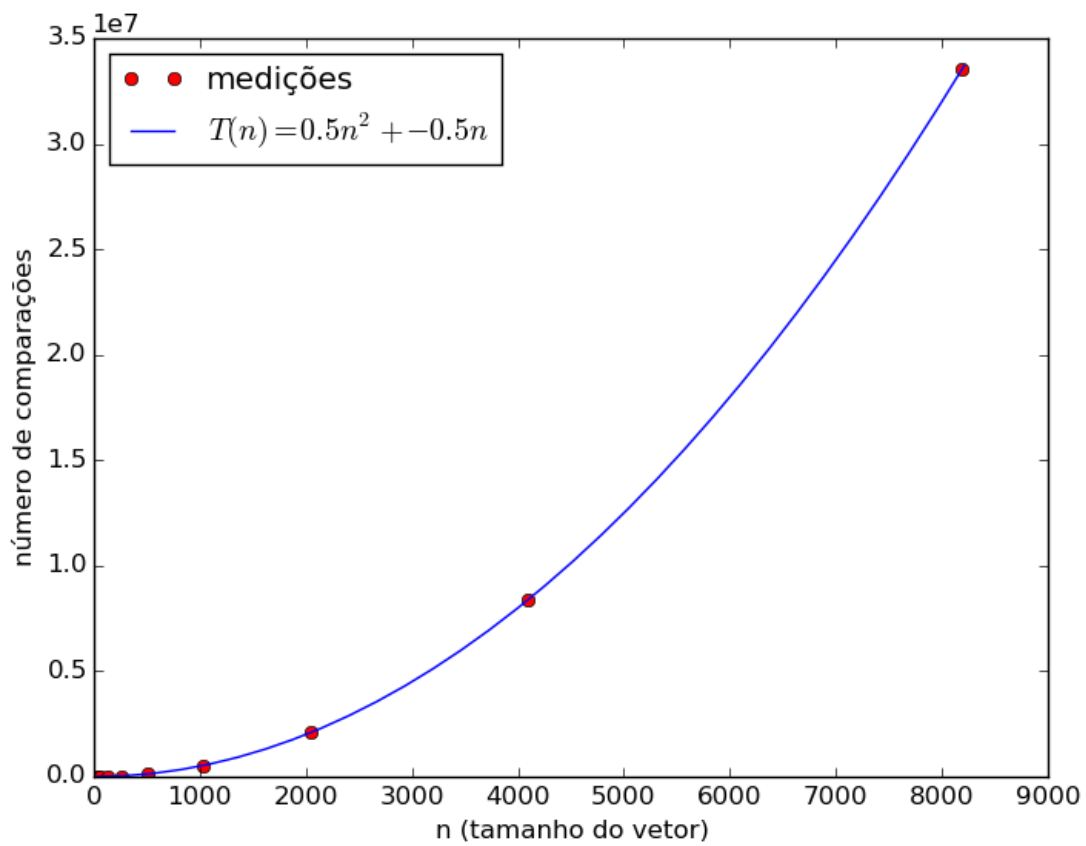


Figura 2.8: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 10%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

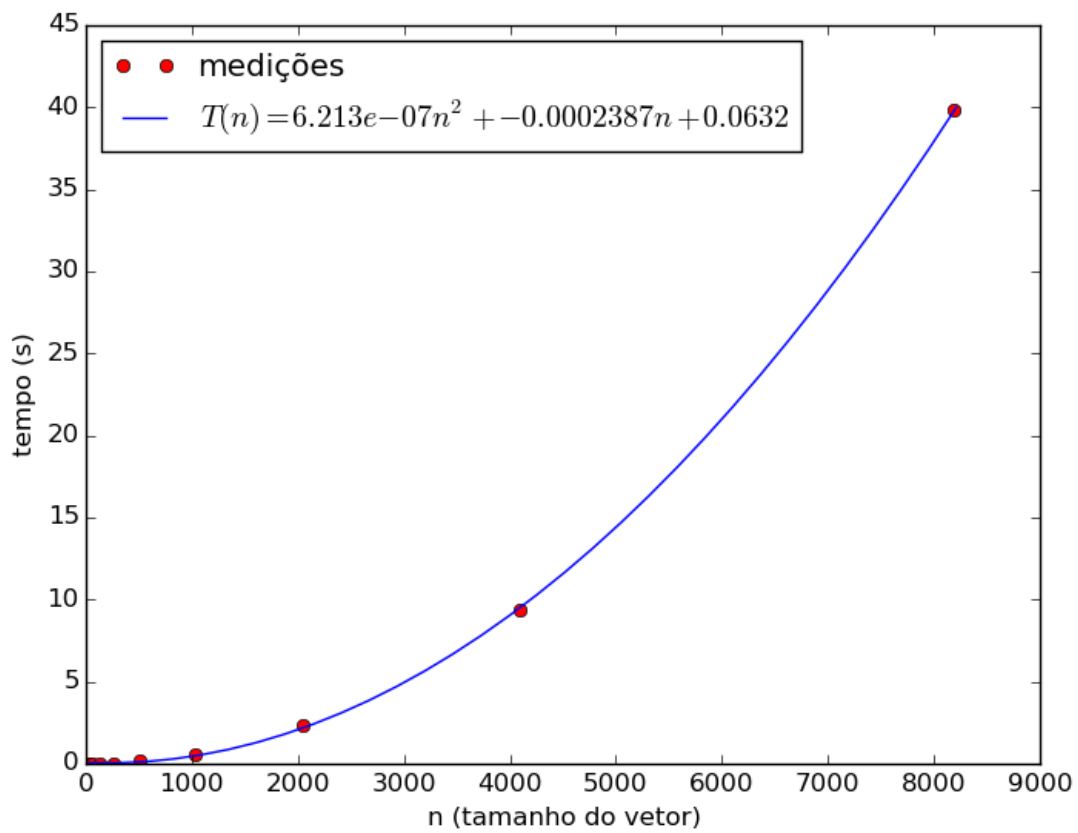


Figura 2.9: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 20%.

Tendo a função $T(n) = 1.639e - 06 * n^2 + 0.001244 * n - 0.3636$ e para o $n = 2^{32}$, $T(2^{32}) = 2.94641 * 10^{302}$

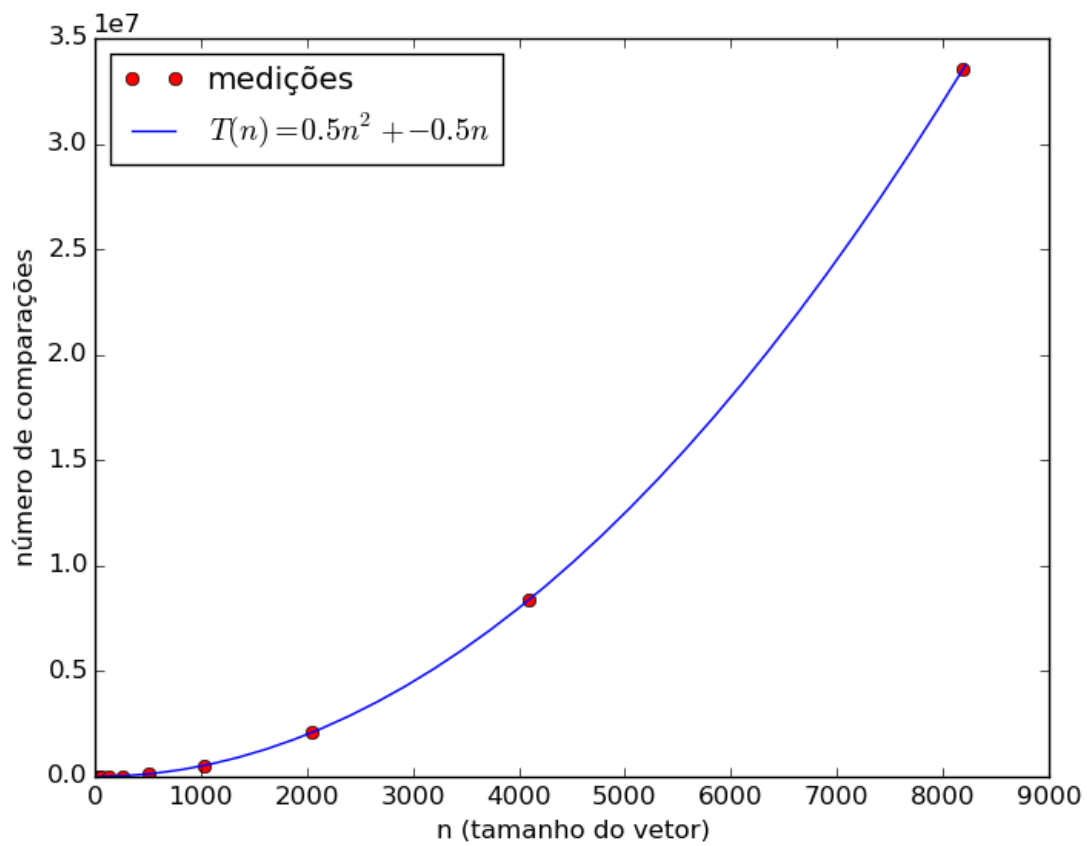


Figura 2.10: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 20%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

n	comparações	tempo(s)
32	496	0.000615
64	2016	0.002372
128	8128	0.009233
256	32640	0.038195
512	130816	0.139211
1024	523776	0.576390
2048	2096128	2.462000
4096	8386560	9.317330
8192	33550336	37.718900

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	0.000651
64	2016	0.002434
128	8128	0.009483
256	32640	0.039904
512	130816	0.153980
1024	523776	0.590784
2048	2096128	2.576600
4096	8386560	9.910330
8192	33550336	41.825800

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

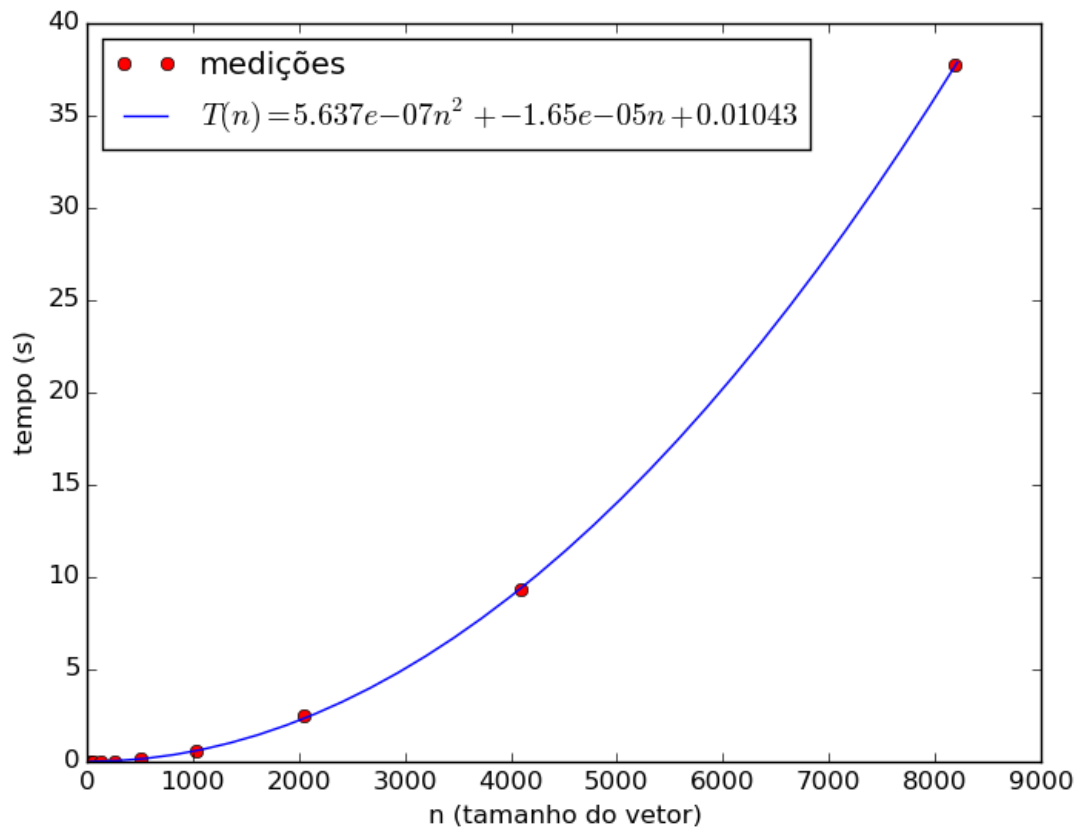


Figura 2.11: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 30%.

Tendo a função $T(n) = 1.305e - 06 * n^2 + 0.0002128 * n - 0.08068$ e para o $n = 2^{32}$, $T(2^{32}) = 2.345989 * 10^{302}$

n	comparações	tempo(s)
32	496	0.000688
64	2016	0.002680
128	8128	0.010606
256	32640	0.040952
512	130816	0.159769
1024	523776	0.623442
2048	2096128	2.730170
4096	8386560	10.305700
8192	33550336	38.857700

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

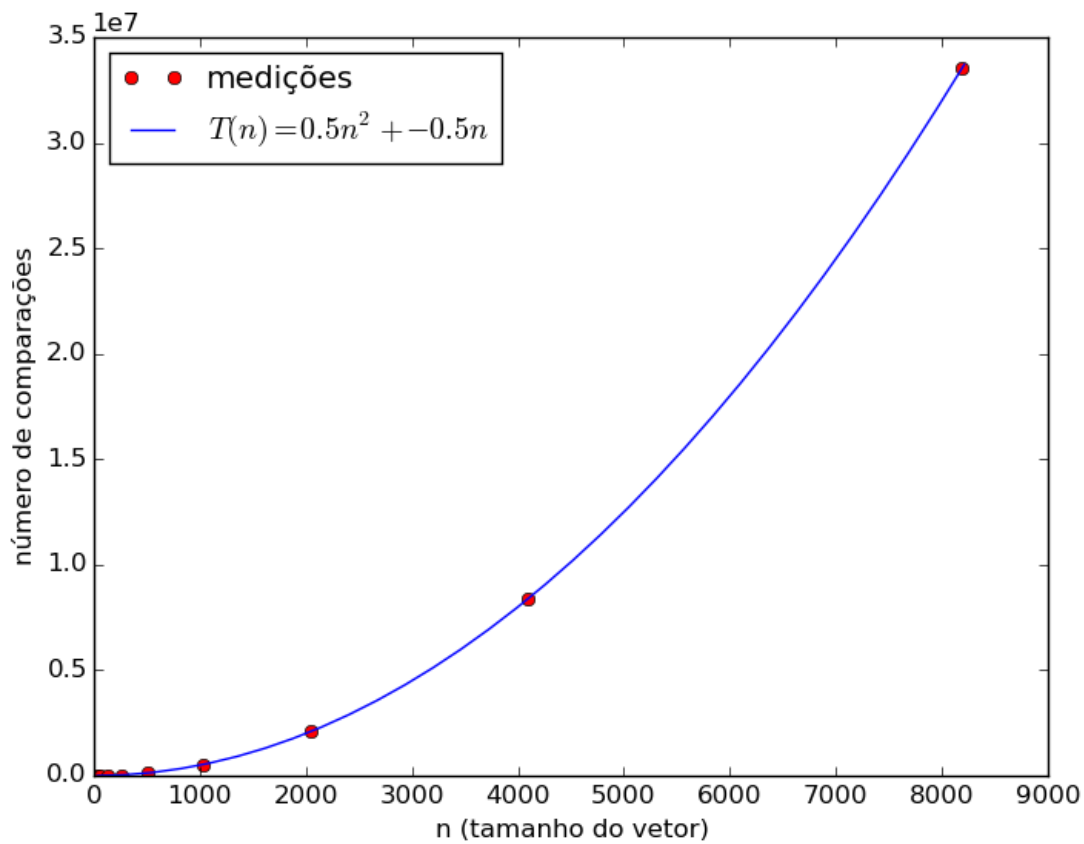


Figura 2.12: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 30%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

n	comparações	tempo(s)
32	496	0.001340
64	2016	0.005808
128	8128	0.020687
256	32640	0.085974
512	130816	0.372805
1024	523776	1.397550
2048	2096128	5.467470
4096	8386560	22.445300
8192	33550336	85.376500

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

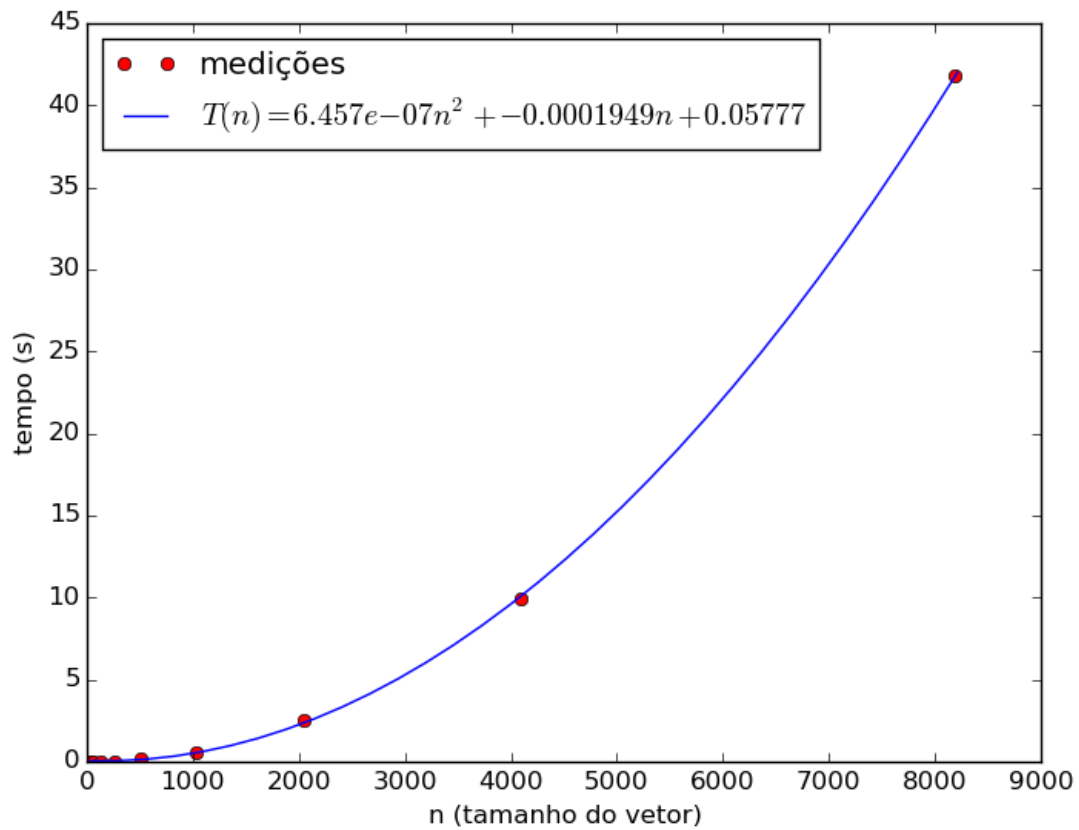


Figura 2.13: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 40%.

Tendo a função $T(n) = 1.425e - 06 * n^2 + 0.0004043 * n - 0.09253$ e para o $n = 2^{32}$, $T(2^{32}) = 2.56171 * 10^{302}$

n	comparações	tempo(s)
32	496	0.001325
64	2016	0.005277
128	8128	0.021163
256	32640	0.088377
512	130816	0.385356
1024	523776	1.373970
2048	2096128	5.488270
4096	8386560	21.906800
8192	33550336	100.340000

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

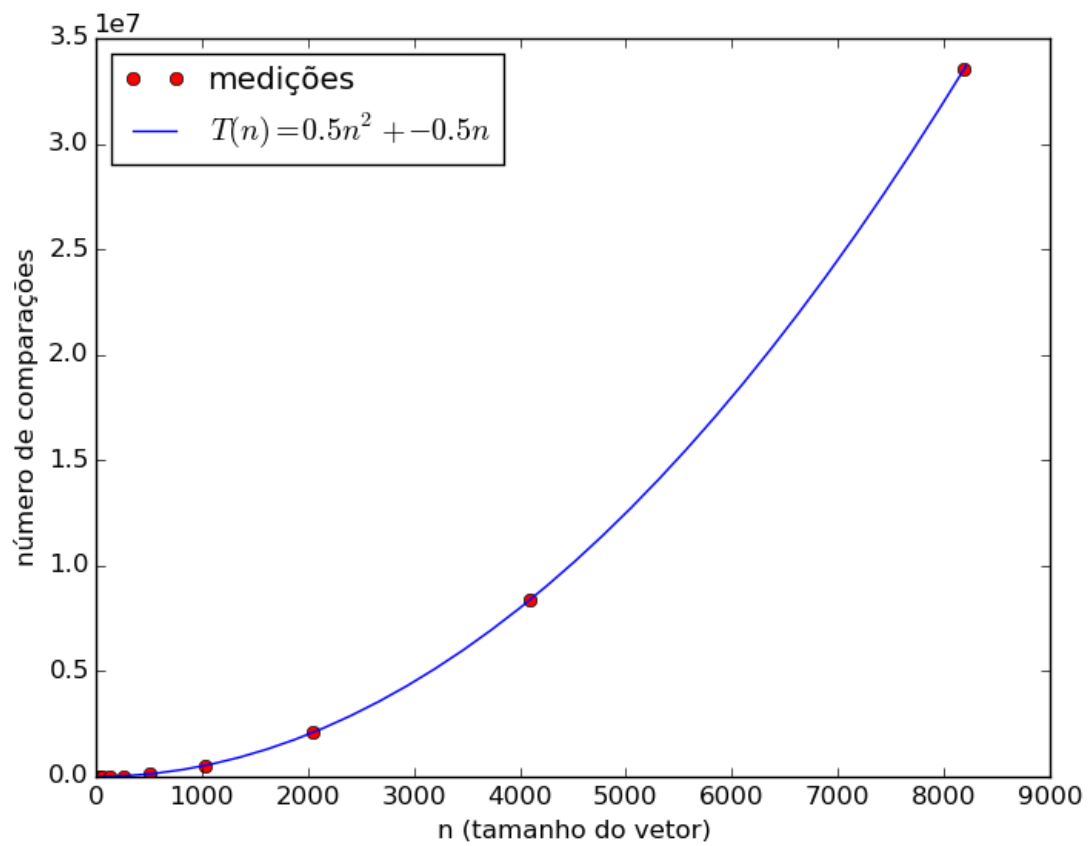


Figura 2.14: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 40%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

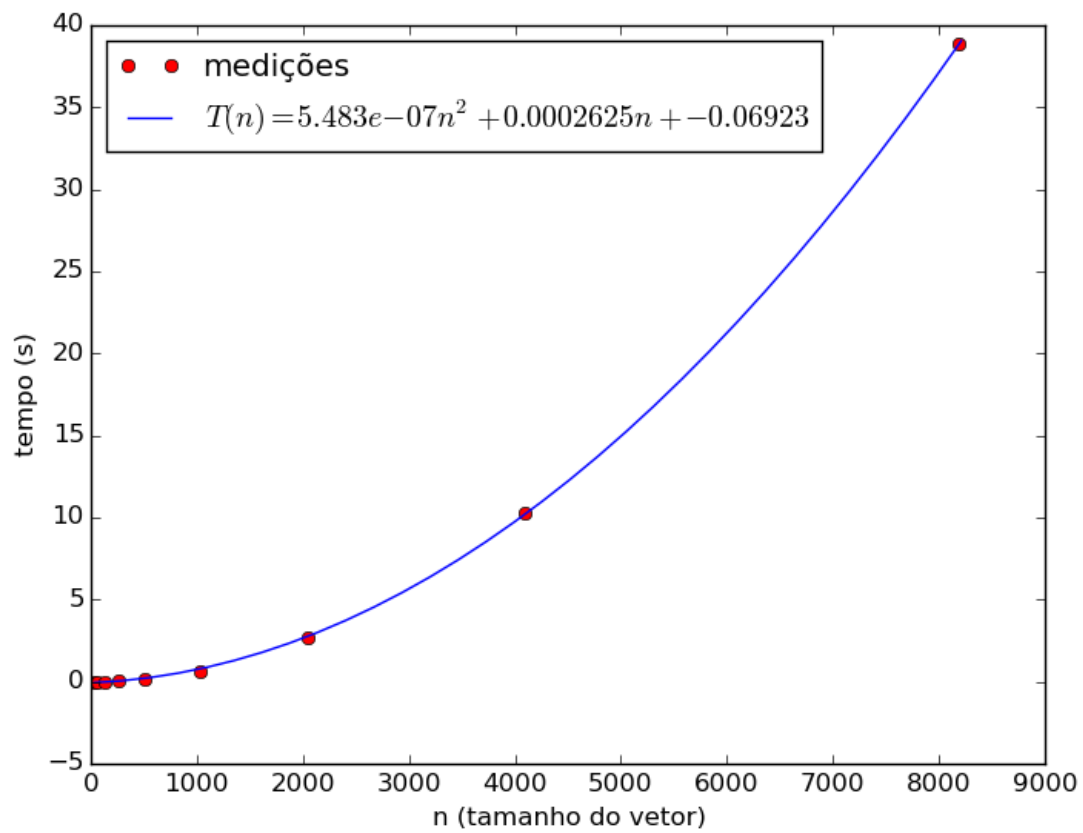


Figura 2.15: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 50%.

Tendo a função $T(n) = 1.211e - 06 * n^2 + 0.0003953 * n - 0.1445$ e para o $n = 2^{32}$, $T(2^{32}) = 2.177006 * 10^{302}$

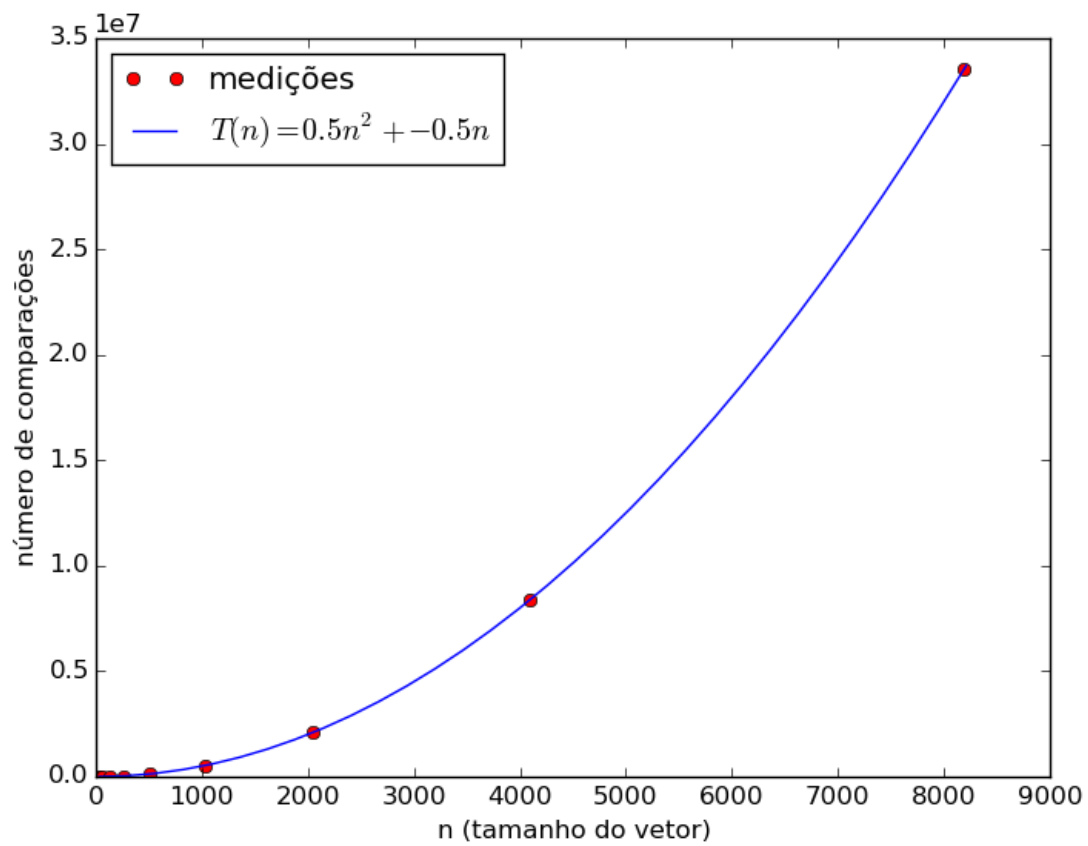


Figura 2.16: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase crescente 50%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

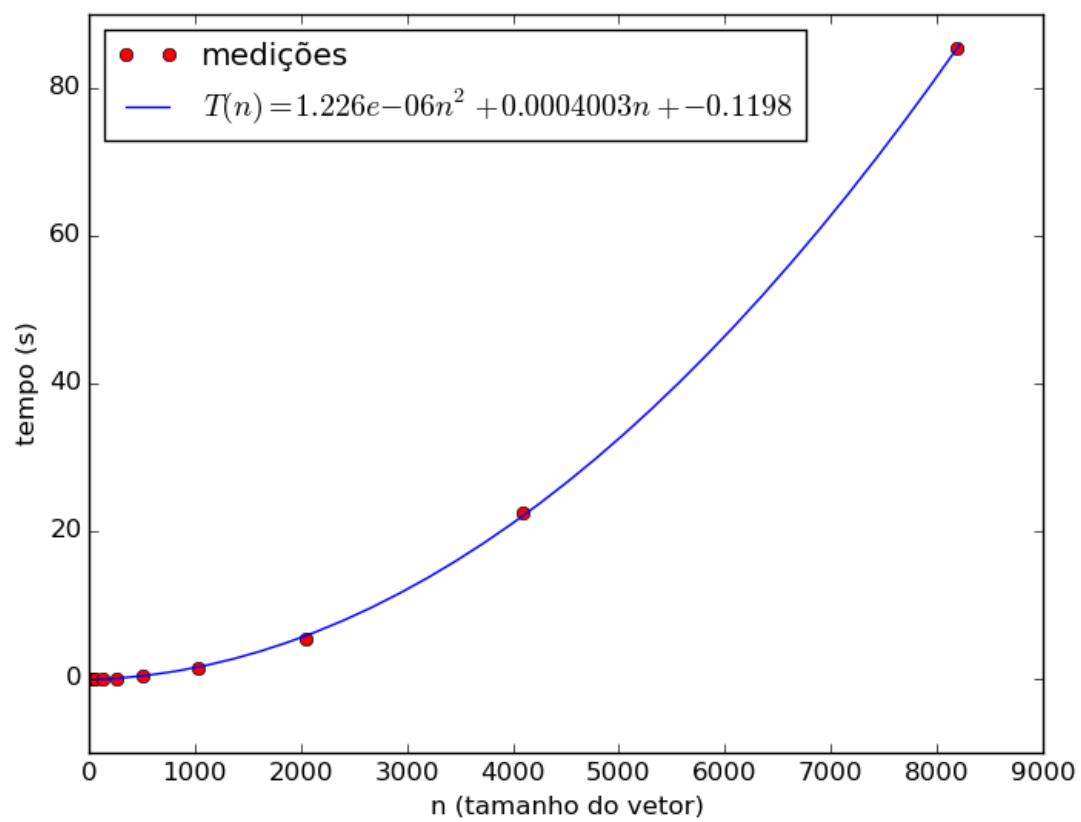


Figura 2.17: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 10%.

Tendo a função $T(n) = 5.404e-07*n^2 - e-05*n - 0.03211$ e para o $n = 2^{32}$, $T(2^{32}) = 9.71473*10^{301}$

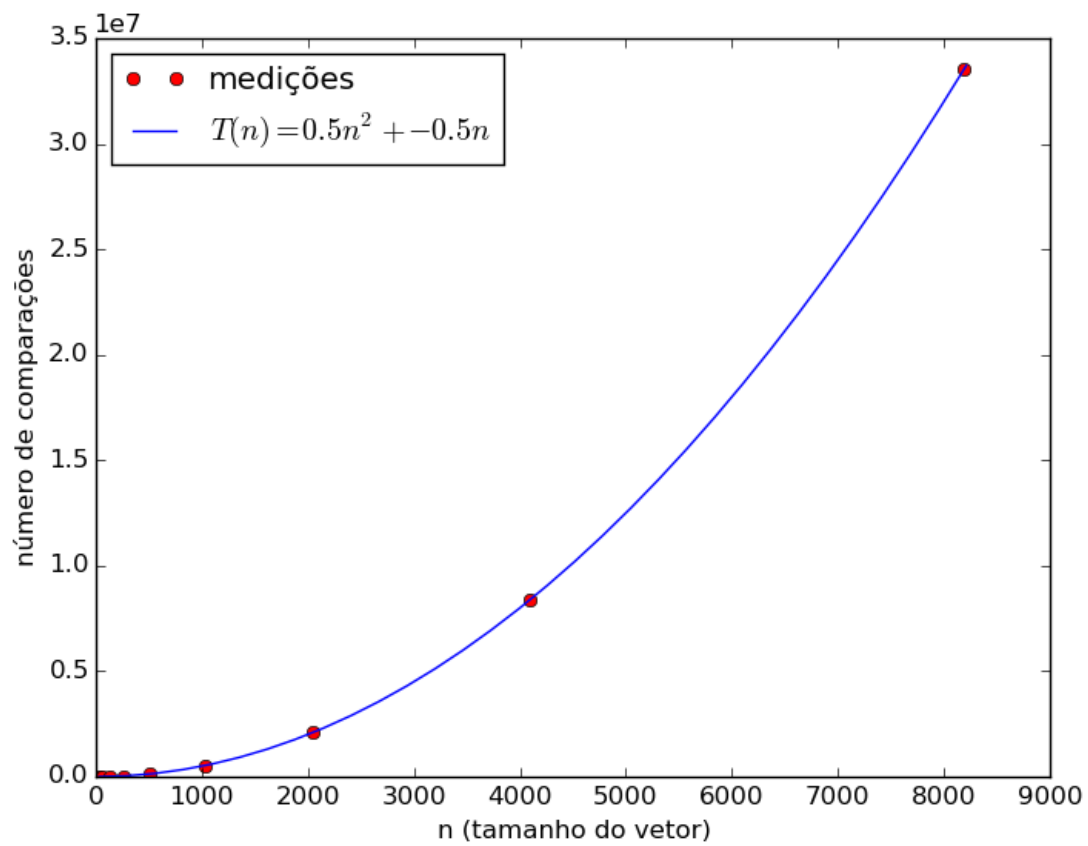


Figura 2.18: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 10%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

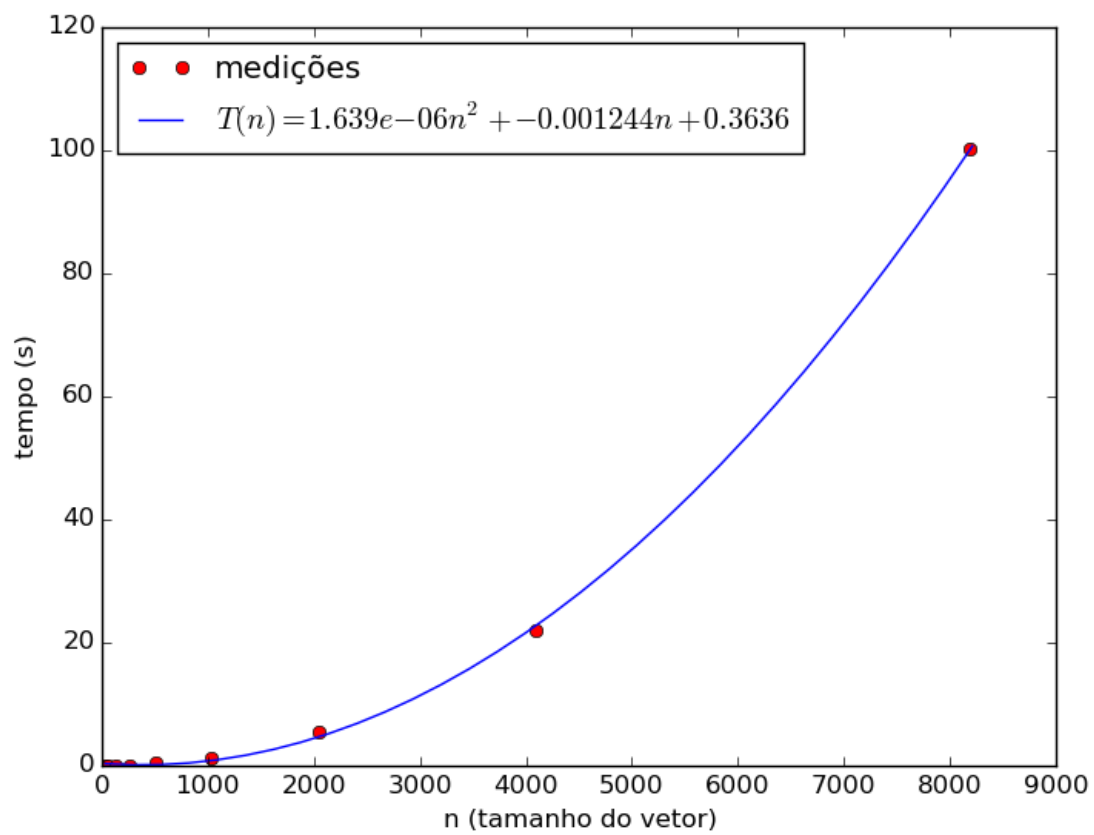


Figura 2.19: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 20%.

Tendo a função $T(n) = 5.404e-07*n^2 - e-05*n - 0.0632$ e para o $n = 2^{32}$, $T(2^{32}) = 1.116906*10^{302}$

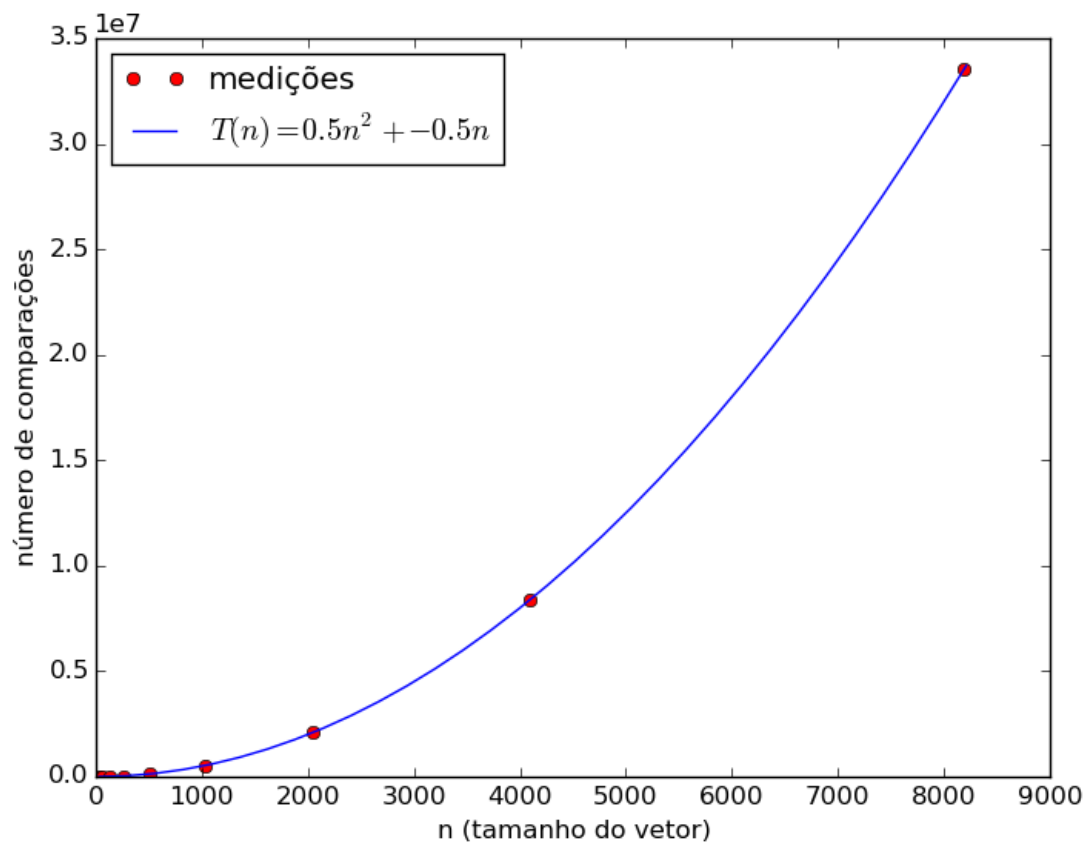


Figura 2.20: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 20%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

n	comparações	tempo(s)
32	496	0.001383
64	2016	0.005260
128	8128	0.022278
256	32640	0.082148
512	130816	0.336932
1024	523776	1.414750
2048	2096128	5.501490
4096	8386560	22.978600
8192	33550336	89.222100

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	0.001275
64	2016	0.005379
128	8128	0.020800
256	32640	0.081763
512	130816	0.329071
1024	523776	1.381690
2048	2096128	5.220150
4096	8386560	22.260100
8192	33550336	92.435900

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

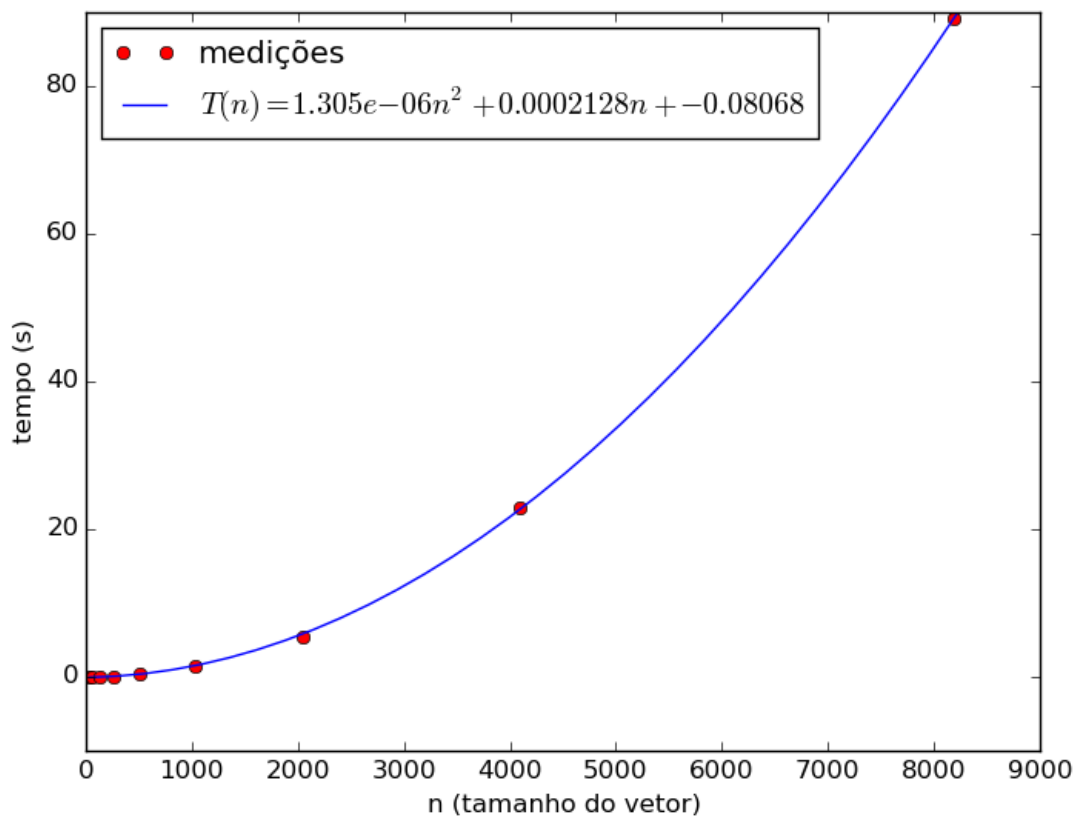


Figura 2.21: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 30%.

Tendo a função $T(n) = 5.637e - 07 * n^2 - 1.65e - 5 * n - 0.01043$ e para o $n = 2^{32}$, $T(2^{32}) = 1.0133596 * 10^{302}$

n	comparações	tempo(s)
32	496	0.001290
64	2016	0.004885
128	8128	0.021722
256	32640	0.085184
512	130816	0.317179
1024	523776	1.226130
2048	2096128	5.382750
4096	8386560	22.192800
8192	33550336	84.292500

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

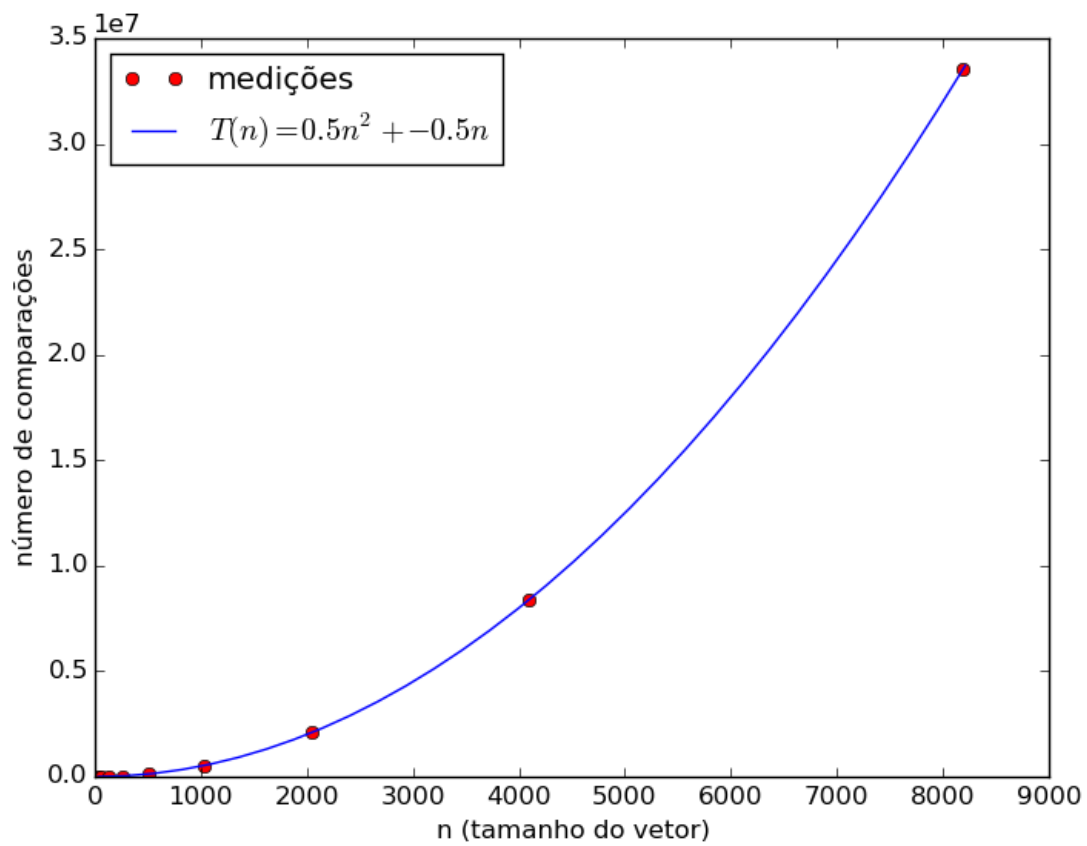


Figura 2.22: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 30%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

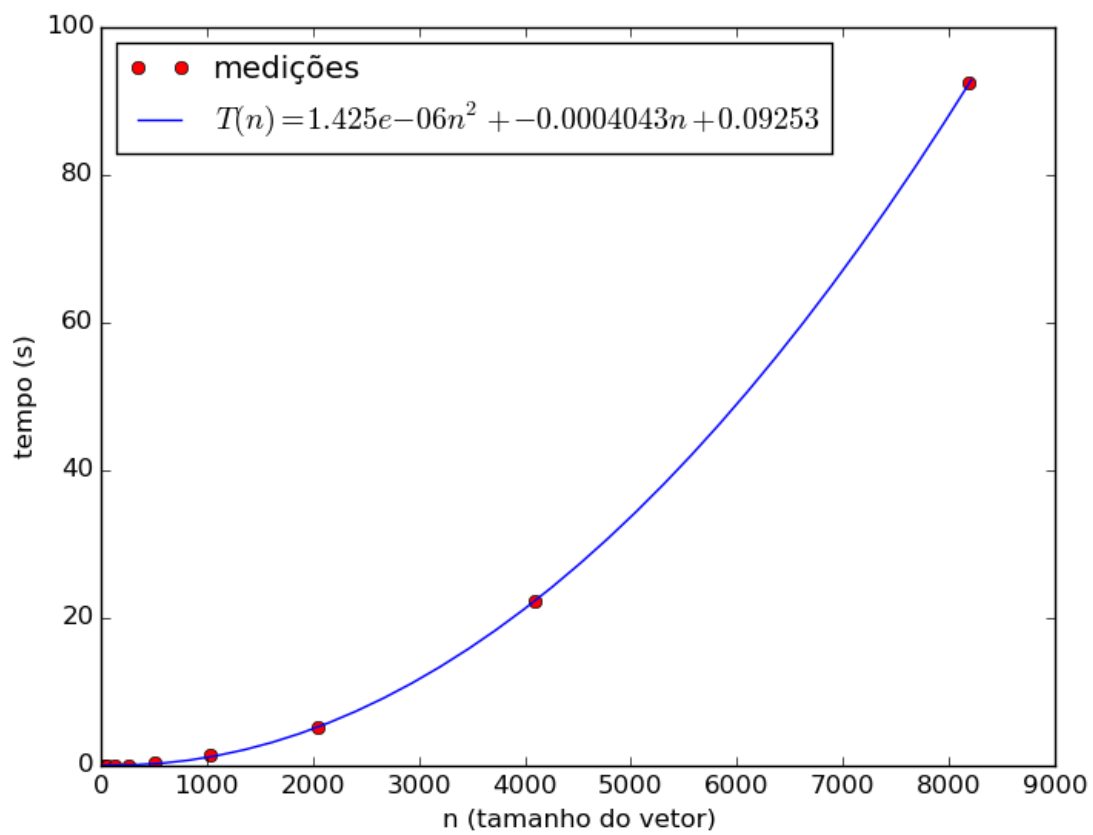


Figura 2.23: A análise do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 40%.

Tendo a função $T(n) = 6.457e-07 * n^2 - 0.00019495 * n - 0.05777$ e para o $n = 2^{32}$, $T(2^{32}) = 1.1607704 * 10^{302}$

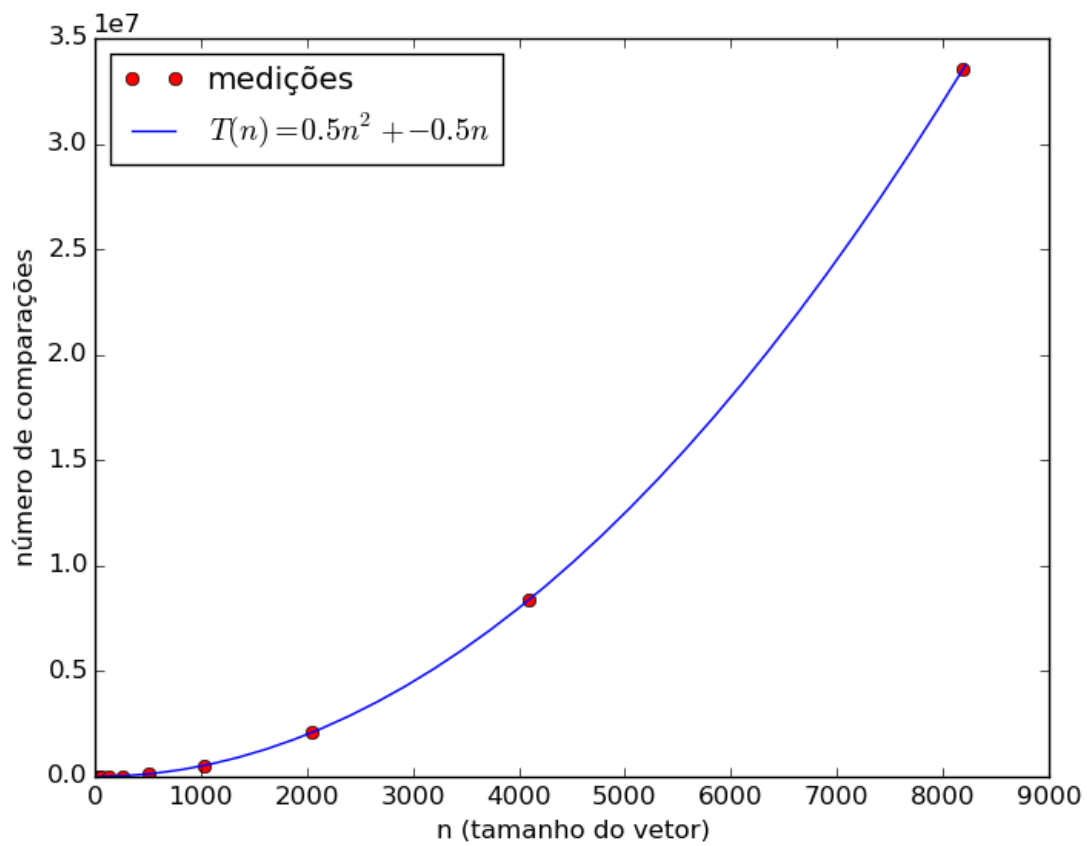


Figura 2.24: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 40%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

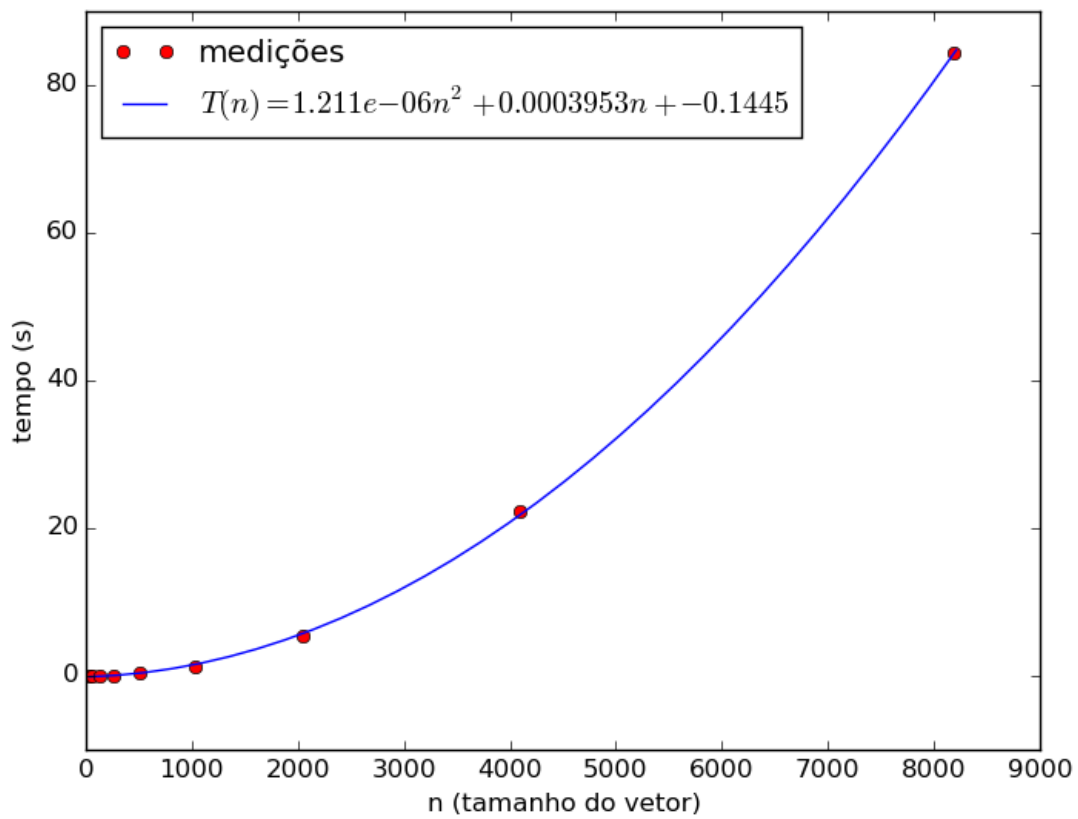


Figura 2.25: A análise do grafico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 50%.

Tendo a função $T(n) = 5.483e-07 * n^2 + 0.0002625 * n - 0.06923$ e para o $n = 2^{32}$, $T(2^{32}) = 9.856751 * 10^{301}$

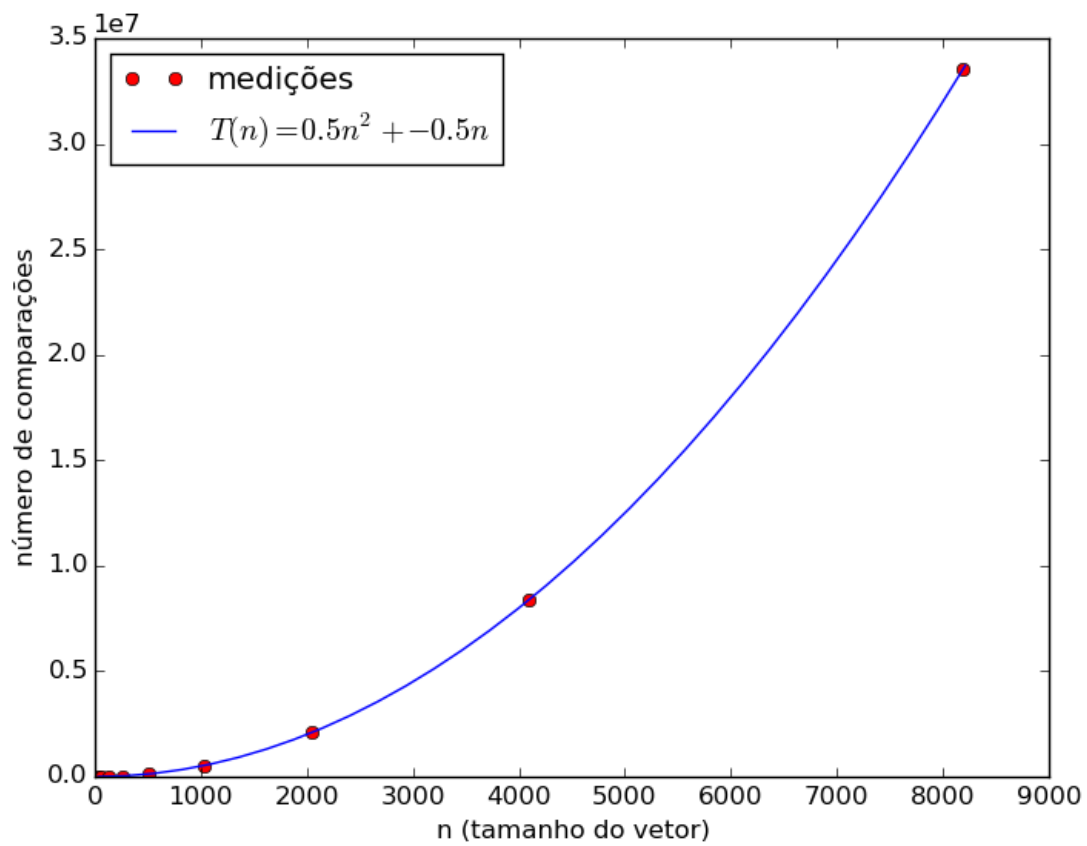


Figura 2.26: A análise em número de comparações do gráfico para 2^{32} segue abaixo para bubblesort de vetor quase decrescente 50%.

Tendo a função $T(n) = 0.5 * n^2 - 0.5 * n$ e para o $n = 2^{32}$, $T(2^{32}) = 8.9884 * 10^{307}$

Apêndice A

Arquivo ../bolha/bolha.py

Listagem A.1: ../bolha/bolha.py

```
1 import numpy as np
2
3 @profile
4 def bubble_sort(a):
5     """ Implementação do método da bolha """
6     for i in range(len(a)):
7         for j in range(len(a)-1-i):
8             if a[j] > a[j+1]:
9                 t = a[j]
10                a[j] = a[j+1]
11                a[j+1] = t
```

Apêndice B

Arquivo ../bolha/ensaio.py

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

```
1 import numpy as np
2 import argparse
3
4 from bolha 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 bubble_sort(vet)
```
