

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

2.1	Bolha com relação ao tempo com vetor aleatório.	8
2.2	Bolha com relação ao número de comparações com vetor aleatório.	9
2.3	Bolha com relação ao tempo com vetor crescente.	11
2.4	Bolha com relação ao número de comparações com vetor crescente.	12
2.5	Bolha com relação ao tempo com vetor decrescente.	14
2.6	Bolha com relação ao número de comparações com vetor decrescente.	15
2.7	bolha com relação ao tempo com vetor quase crescente 10%.	17
2.8	Bolha com relação ao número de comparações com vetor quase crescente 10%.	18
2.9	Bolha com relação ao tempo com vetor quase crescente 20%.	20
2.10	Bolha com relação ao número de comparações com vetor quase crescente 20%.	21
2.11	Bolha com relação ao tempo com vetor quase crescente 30%.	23
2.12	Bolha com relação ao número de comparações com vetor quase crescente 30%.	24
2.13	Bolha com relação ao tempo com vetor quase crescente 40%.	26
2.14	Bolha com relação ao número de comparações com vetor quase crescente 40%.	27
2.15	Bolha com relação ao tempo com vetor quase crescente 50%.	29
2.16	Bolha com relação ao número de comparações com vetor quase crescente 50%.	30
2.17	Bolha com relação ao tempo com vetor quase decrescente 10%.	32
2.18	Bolha com relação ao número de comparações com vetor quase decrescente 10%.	33
2.19	Bolha com relação ao tempo com vetor quase decrescente 20%.	35
2.20	Bolha com relação ao número de comparações com vetor quase decrescente 20%.	36
2.21	Bolha com relação ao tempo com vetor quase decrescente 30%.	38
2.22	Bolha com relação ao número de comparações com vetor quase decrescente 30%.	39
2.23	Bolha com relação ao tempo com vetor quase decrescente 40%.	41
2.24	Bolha com relação ao número de comparações com vetor quase decrescente 40%.	42
2.25	Bolha com relação ao tempo com vetor quase decrescente 50%.	44
2.26	Bolha com relação ao número de comparações com vetor quase decrescente 50%.	45

Lista de Tabelas

2.1	Bolha com Vetores Aleatorio	7
2.2	Bolha com Vetores Crescentes	10
2.3	Bolha com Vetores Decrescentes	13
2.4	Bolha com Vetores Quase Crescentes 10%	16
2.5	Bolha com Vetores Quase Crescentes 20%	19
2.6	Bolha com Vetores Quase Crescentes 30%	22
2.7	Bolha com Vetores Quase Crescentes 40%	25
2.8	Bolha com Vetores Quase Crescentes 50%	28
2.9	Bolha com Vetores Quase Decrescentes 10%	31
2.10	Bolha com Vetores Quase Decrescentes 20%	34
2.11	Bolha com Vetores Quase Decrescentes 30%	37
2.12	Bolha com Vetores Quase Decrescentes 40%	40
2.13	Bolha com Vetores Quase Decrescentes 50%	43

Lista de Listagens

A.1	../bolha/bolha.py	46
B.1	../bolha/ensaio.py	47

Sumário

Lista de Figuras	2
Lista de Tabelas	3
1 Análise	6
2 Resultados	7
2.1 Tabelas	7
 Apêndice	 46
A Arquivo ../bolha/bolha.py	46
B Arquivo ../bolha/ensaio.py	47

Capítulo 1

Análise

O algoritmo Bubble Sort utiliza uma técnica básica, na qual várias passagens são feitas de forma que em cada uma delas, comparam-se dois elementos adjacentes e caso o segundo elemento for menor que o primeiro, então suas posições são trocadas. Isso é feito até que nenhum elemento tenha sido trocado de posição na passagem anterior, ou seja, até os elementos estarem todos ordenados.

As vantagens desse algoritmo são que é bem simples de implementar e é considerado estável, sendo recomendado para vetores quase ordenados. Por outro lado, é um algoritmo lento, principalmente se o tamanho do vetor for grande, devido ao fato de fazer muitas comparações.

No melhor caso, o algoritmo executa um número de operações linear, $teta(n)$, onde n representa o número de elementos do vetor. No caso médio e no pior caso, são feitas $teta(n^2)$ operações. Dessa forma, a complexidade do Bubble Sort é de ordem quadrática. Por isso, ele não é recomendado para programas que precisem de velocidade e operem com quantidade elevada de dados.

Em termos de memória, o Bubble Sort tem complexidade de espaço constante, $teta(1)$.

Capítulo 2

Resultados

2.1 Tabelas

n	comparações	tempo(s)
32	496	0.001244
64	2016	0.004338
128	8128	0.018608
256	32640	0.070951
512	130816	0.270437
1024	523776	1.226090
2048	2096128	4.880630
4096	8386560	18.790900
8192	33550336	72.747300

Tabela 2.1: *Bolha com Vetores Aleatorio*

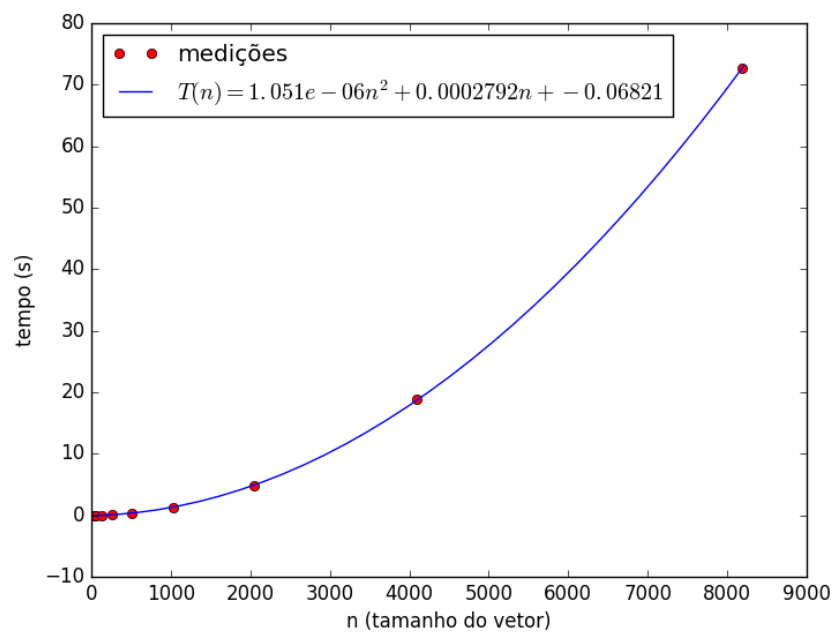


Figura 2.1: Bolha com relação ao tempo com vetor aleatório.

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

$$1.51 * 10^{-6} * n^2 + 0.0002792 * n - 0.06821 = 2.785458475e + 13$$

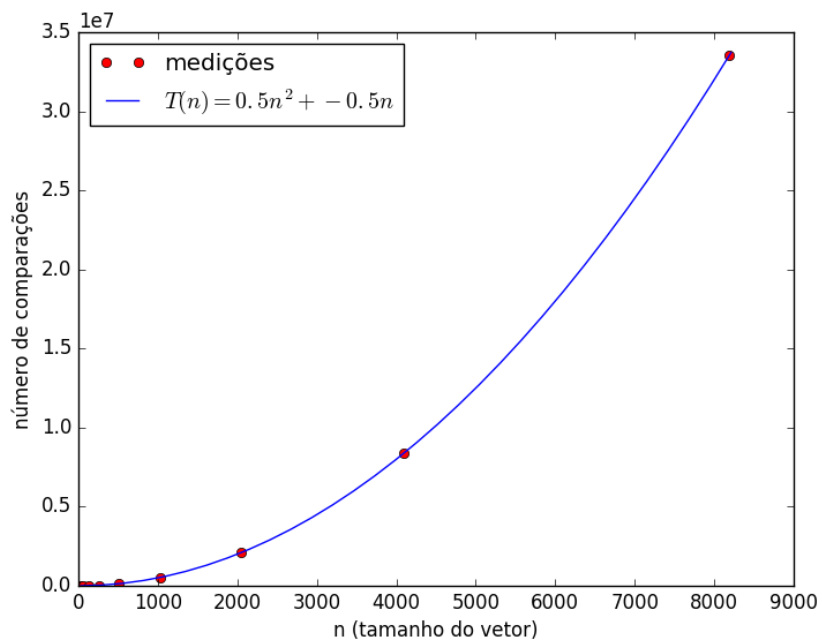


Figura 2.2: Bolha com relação ao número de comparações com vetor aleatório.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000738
64	2016	0.002399
128	8128	0.009983
256	32640	0.040741
512	130816	0.157150
1024	523776	0.645521
2048	2096128	2.723680
4096	8386560	10.420300
8192	33550336	43.467400

Tabela 2.2: *Bolha com Vetores Crescentes*

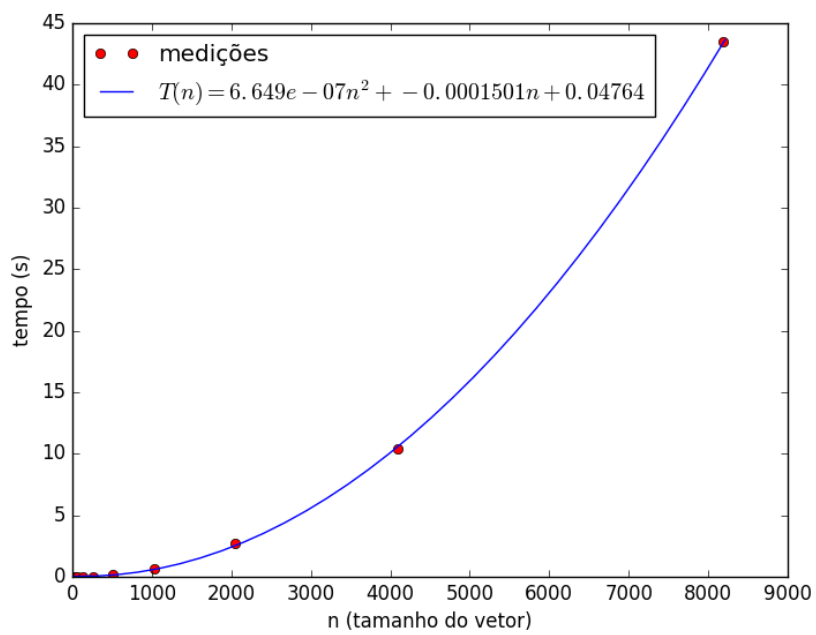


Figura 2.3: Bolha com relação ao tempo com vetor crescente.

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

$$6.649 * 10^{-7} * n^2 - 0.0001501 * n - 0.04764 = 1.2265239e + 13$$

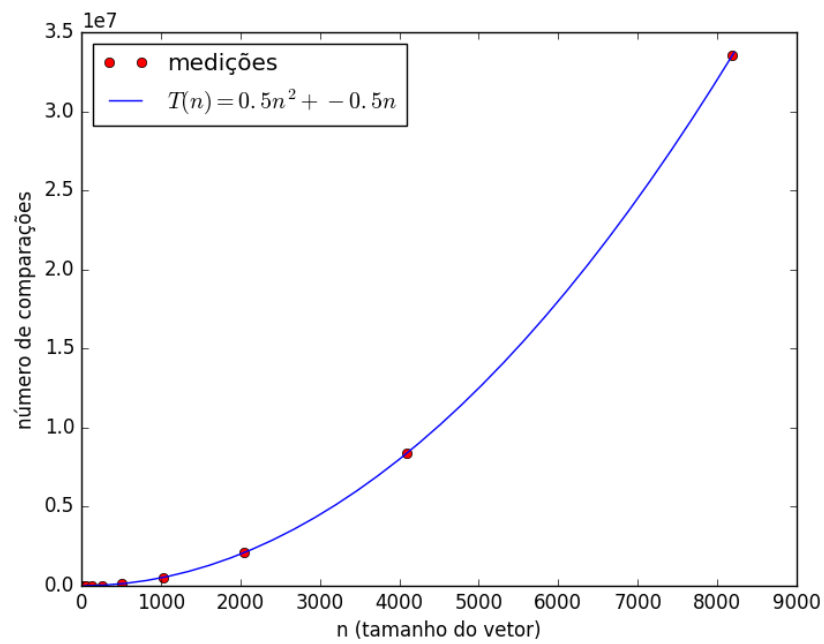


Figura 2.4: Bolha com relação ao número de comparações com vetor crescente.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001707
64	2016	0.006611
128	8128	0.025607
256	32640	0.096211
512	130816	0.447886
1024	523776	1.752420
2048	2096128	6.261640
4096	8386560	26.895500
8192	33550336	111.340000

Tabela 2.3: *Bolha com Vetores Decrescentes*

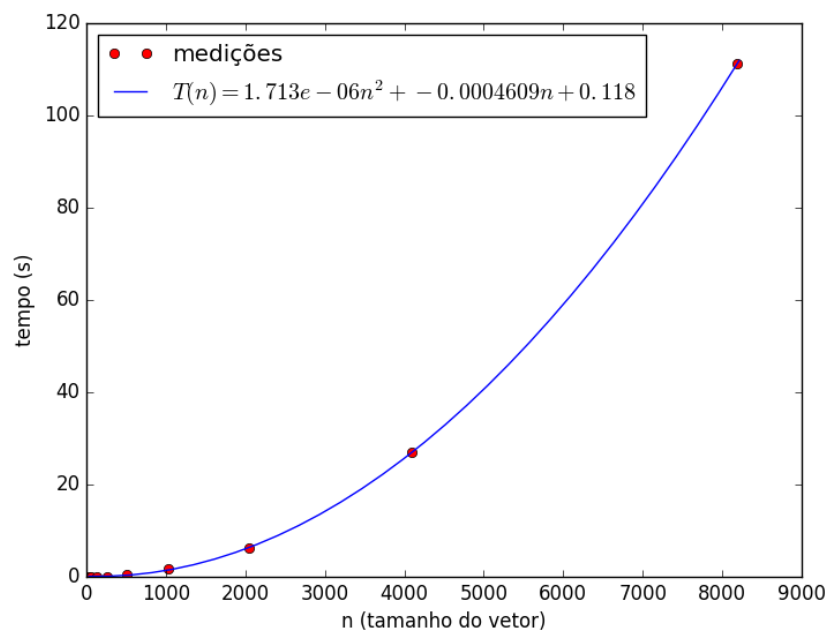


Figura 2.5: Bolha com relação ao tempo com vetor decrescente.

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

$$1.713 * 10^{-6} * n^2 - 0.0004609 * n + 0.118 = 3.159927062e + 13$$

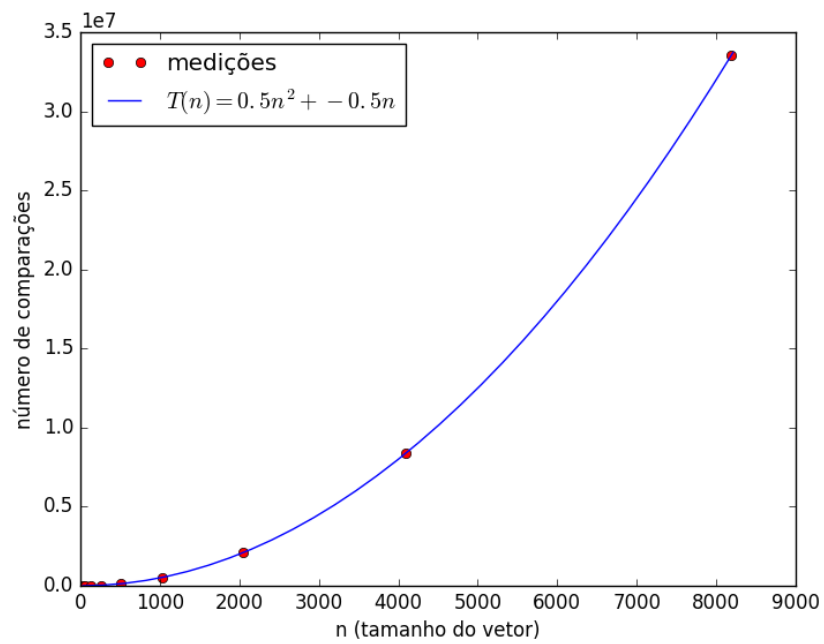


Figura 2.6: Bolha com relação ao número de comparações com vetor decrescente.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000647
64	2016	0.002479
128	8128	0.010025
256	32640	0.036940
512	130816	0.168592
1024	523776	0.715718
2048	2096128	2.897700
4096	8386560	10.558300
8192	33550336	44.095600

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

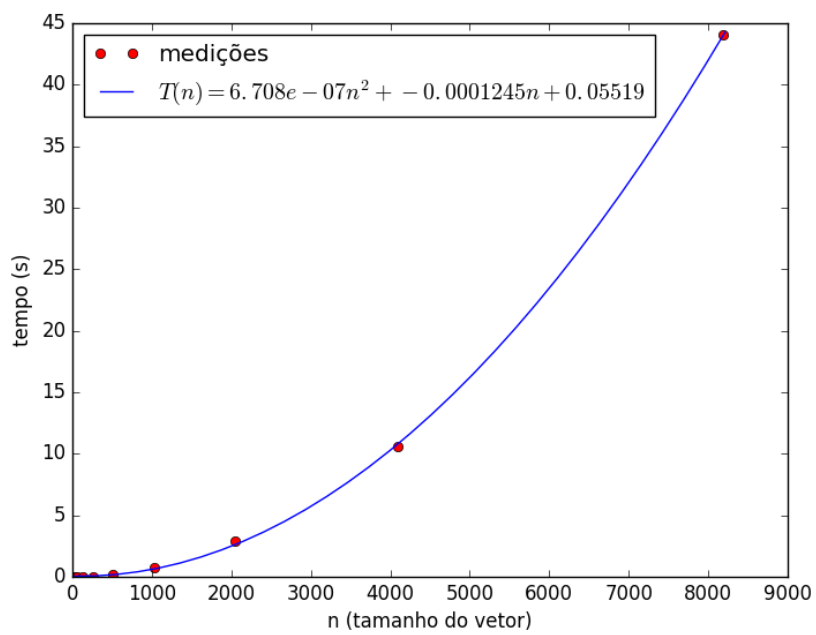


Figura 2.7: bolha com relação ao tempo com vetor quase crescente 10%.

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

$$6.708 * 10^{-7} * n^2 - 0.0001245 * n + 0.05519 = 1.237407539e + 13$$

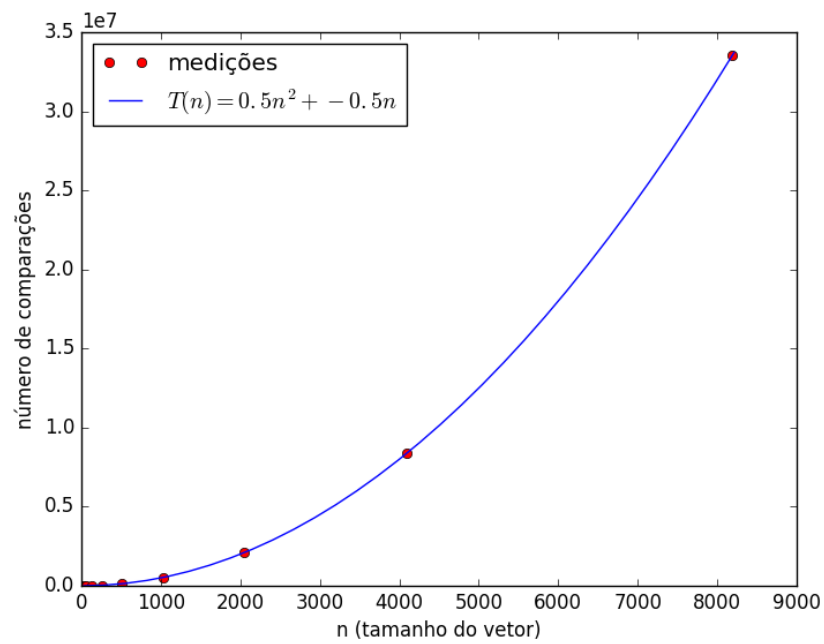


Figura 2.8: Bolha com relação ao número de comparações com vetor quase crescente 10%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000688
64	2016	0.002663
128	8128	0.011074
256	32640	0.043802
512	130816	0.167792
1024	523776	0.622394
2048	2096128	2.595740
4096	8386560	10.874400
8192	33550336	41.212800

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

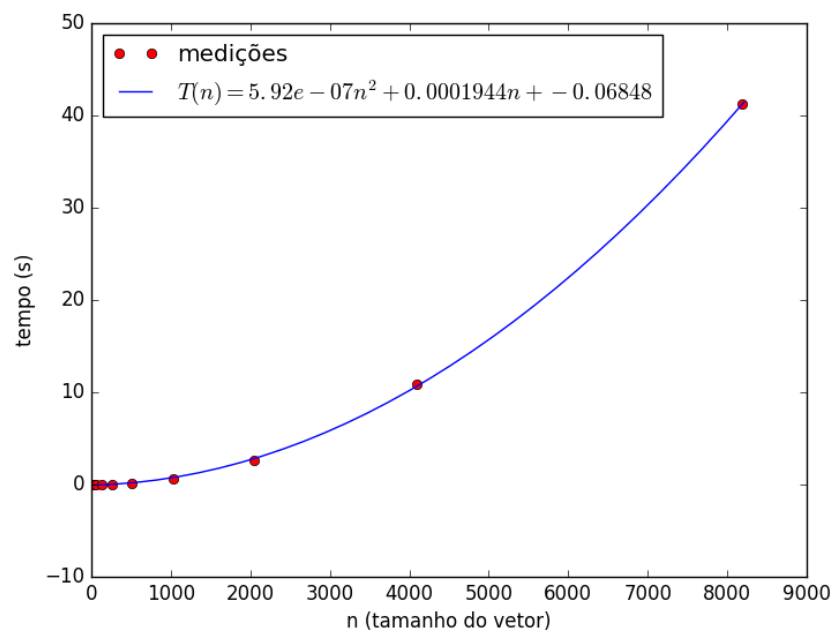


Figura 2.9: Bolha com relação ao tempo com vetor quase crescente 20%.

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

$$5.92 * 10^{-7} * n^2 - 0.0001944 * n - 0.06848 = 1.092047166e + 13$$

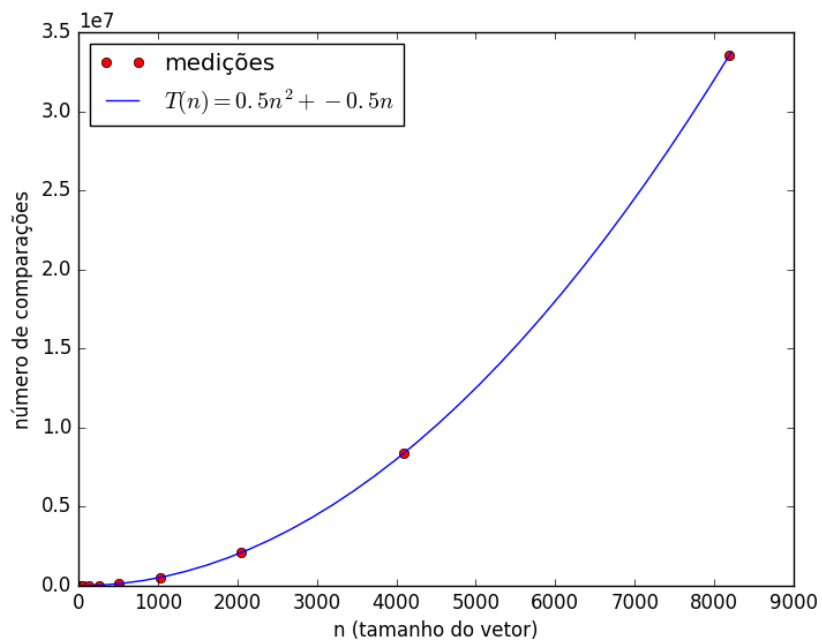


Figura 2.10: Bolha com relação ao número de comparações com vetor quase crescente 20%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000769
64	2016	0.002671
128	8128	0.011554
256	32640	0.041458
512	130816	0.164987
1024	523776	0.655907
2048	2096128	2.865800
4096	8386560	11.818100
8192	33550336	47.561800

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

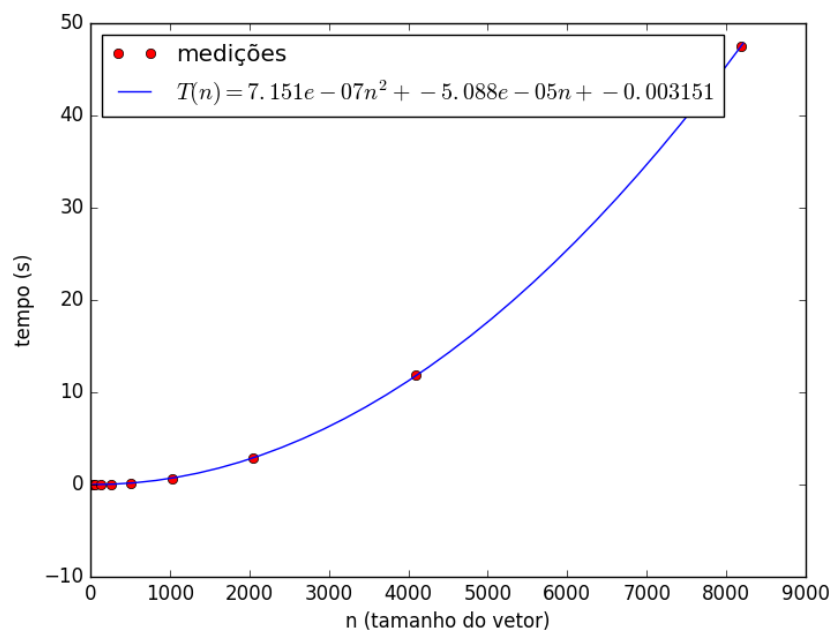


Figura 2.11: Bolha com relação ao tempo com vetor quase crescente 30%.

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

$$7.151 * 10^{-7} * n^2 - 5.088 * 10^{-5} * n - 0.003151 = 1.319126647e + 13$$

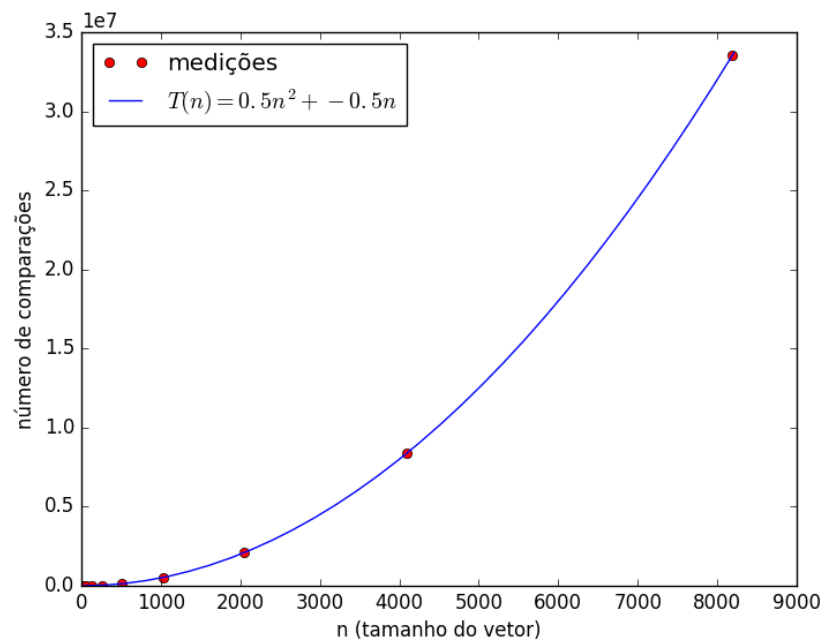


Figura 2.12: Bolha com relação ao número de comparações com vetor quase crescente 30%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000690
64	2016	0.002716
128	8128	0.010886
256	32640	0.044047
512	130816	0.190109
1024	523776	0.719628
2048	2096128	2.839420
4096	8386560	11.083800
8192	33550336	44.255100

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

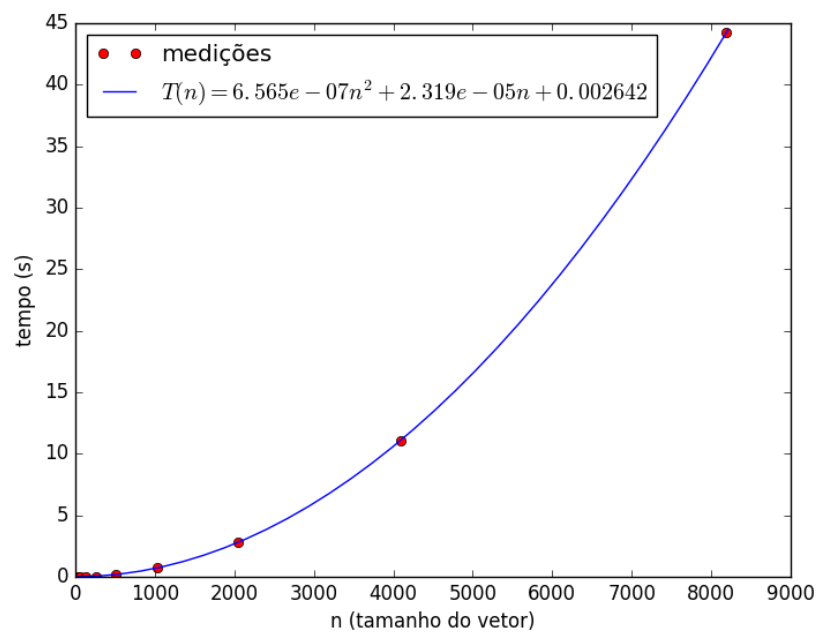


Figura 2.13: Bolha com relação ao tempo com vetor quase crescente 40%.

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

$$6.565 * 10^{-7} * n^2 + 2.319 * 10^{-5} * n + 0.002642 = 1.211028758e + 13$$

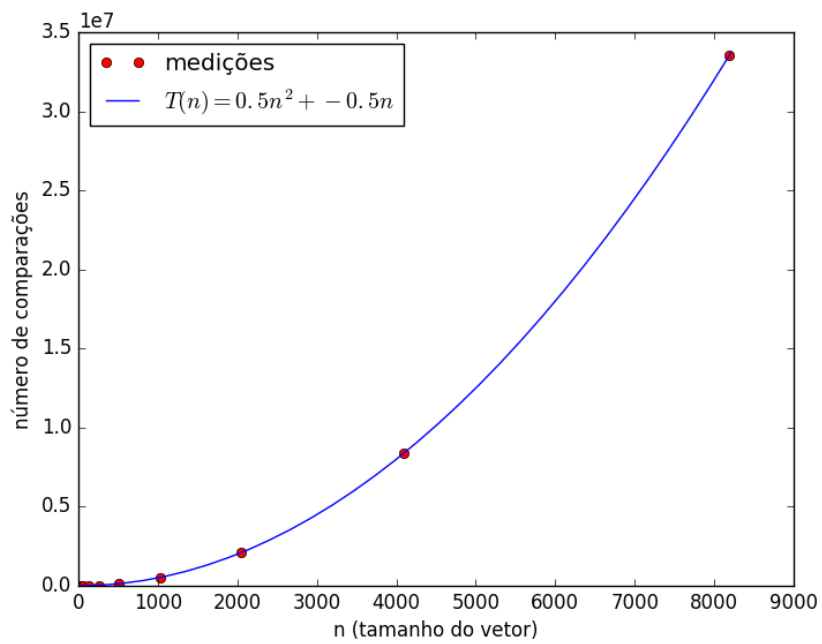


Figura 2.14: Bolha com relação ao número de comparações com vetor quase crescente 40%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.000776
64	2016	0.003128
128	8128	0.012131
256	32640	0.044830
512	130816	0.181569
1024	523776	0.748040
2048	2096128	3.134900
4096	8386560	13.150300
8192	33550336	49.897100

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

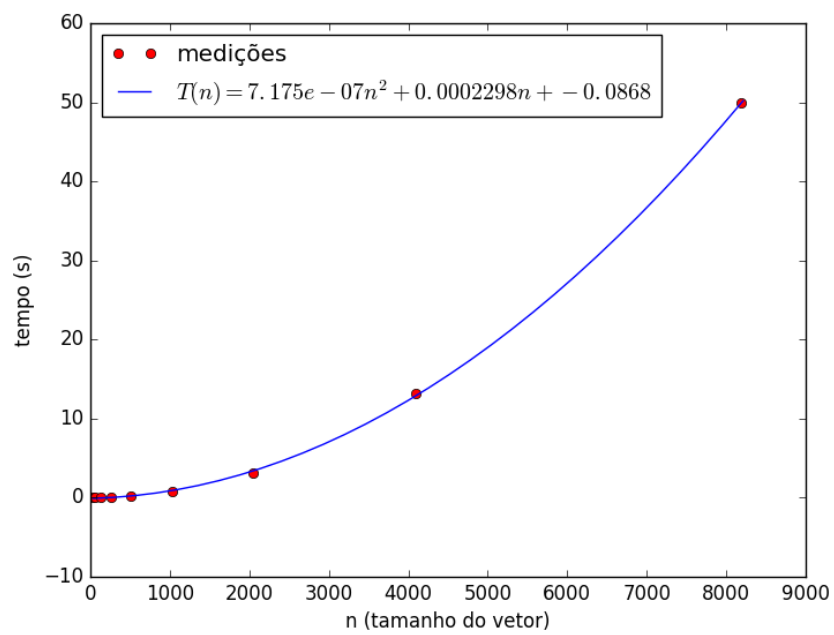


Figura 2.15: Bolha com relação ao tempo com vetor quase crescente 50%.

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

$$7.151 * 10^{-7} * n^2 + 0.0002298 * n - 0.0868 = 1.319126767e + 13$$

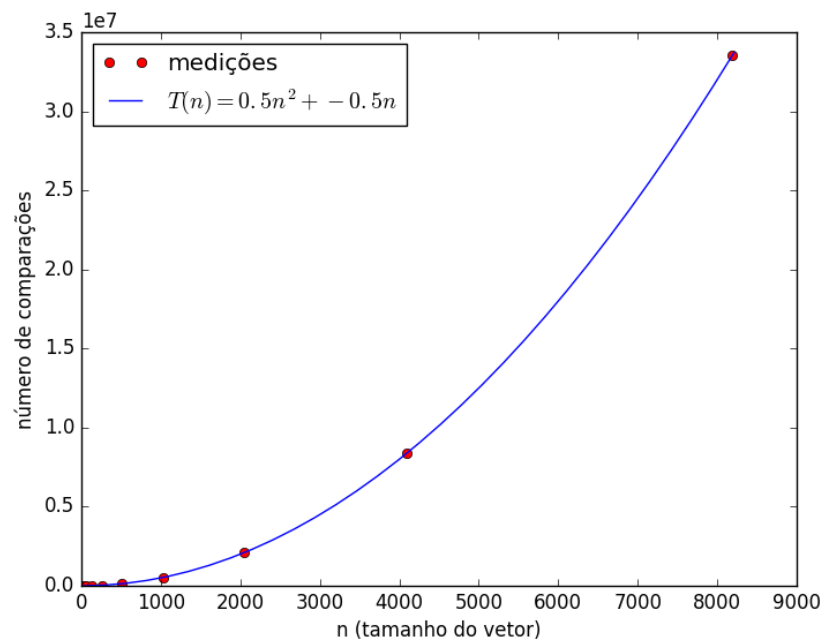


Figura 2.16: Bolha com relação ao número de comparações com vetor quase crescente 50%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001590
64	2016	0.006128
128	8128	0.025919
256	32640	0.099607
512	130816	0.429297
1024	523776	1.960880
2048	2096128	6.783710
4096	8386560	28.090500
8192	33550336	116.388000

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

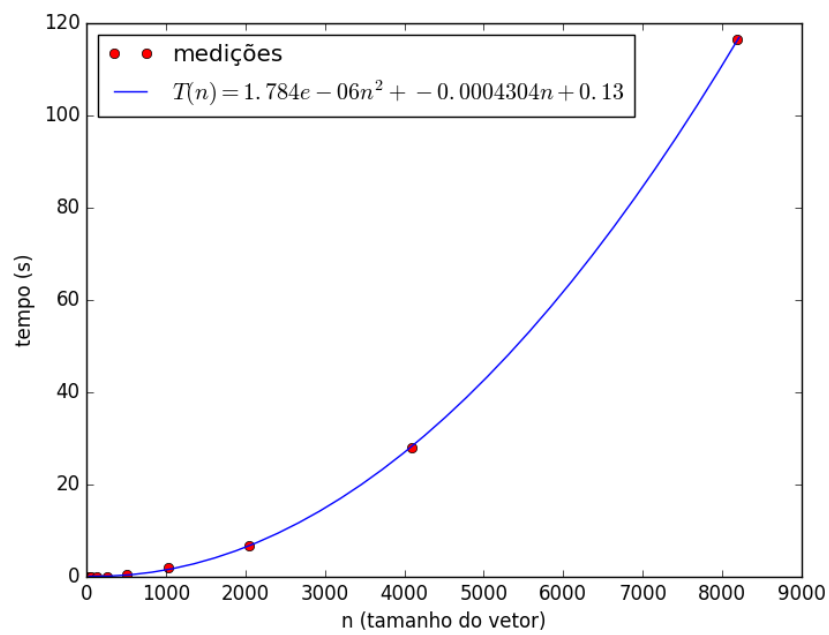


Figura 2.17: Bolha com relação ao tempo com vetor quase decrescente 10%.

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

$$1.784 * 10^{-6} * n^2 - 0.0004304 * n + 0.13 = 3.290898958e + 13$$

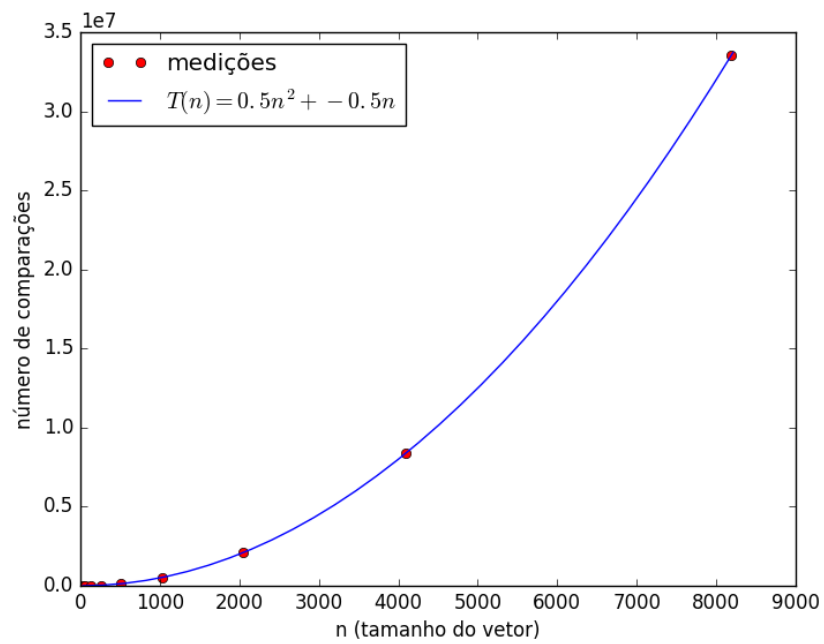


Figura 2.18: Bolha com relação ao número de comparações com vetor quase decrescente 10%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001617
64	2016	0.006157
128	8128	0.025687
256	32640	0.097898
512	130816	0.408679
1024	523776	1.560380
2048	2096128	6.813210
4096	8386560	26.124200
8192	33550336	100.624000

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

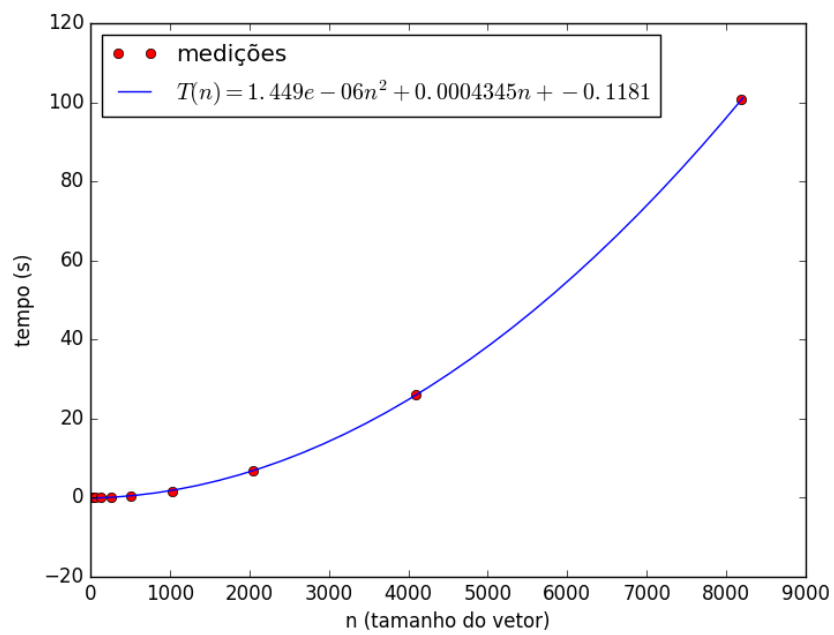


Figura 2.19: Bolha com relação ao tempo com vetor quase decrescente 20%.

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

$$1.449 * 10^{-6} * n^2 - 0.0004345 * n - 0.1181 = 2.67293303e + 13$$

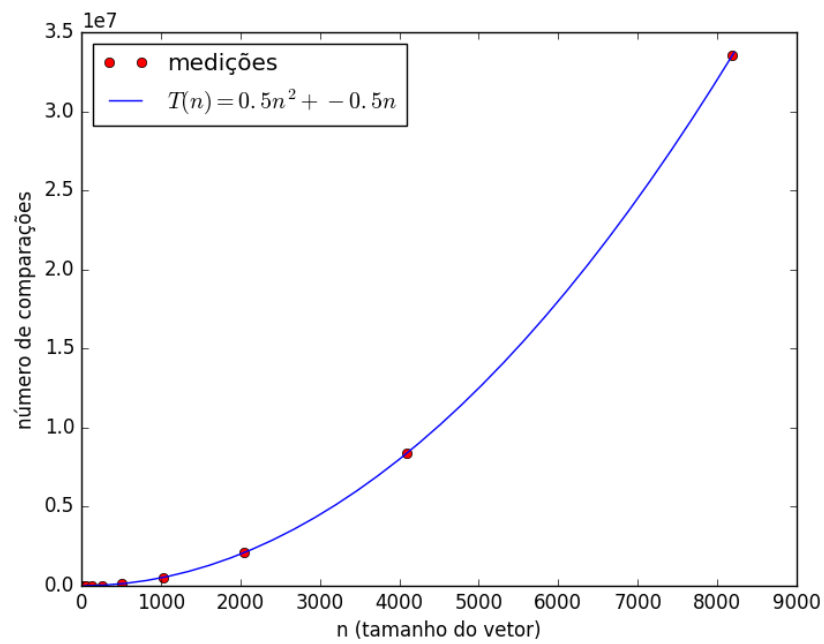


Figura 2.20: Bolha com relação ao número de comparações com vetor quase decrescente 20%.

Análise com relação ao tamanho do vetor 2³²:

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001714
64	2016	0.006240
128	8128	0.024473
256	32640	0.096080
512	130816	0.391649
1024	523776	1.766420
2048	2096128	6.900400
4096	8386560	24.923900
8192	33550336	100.544000

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

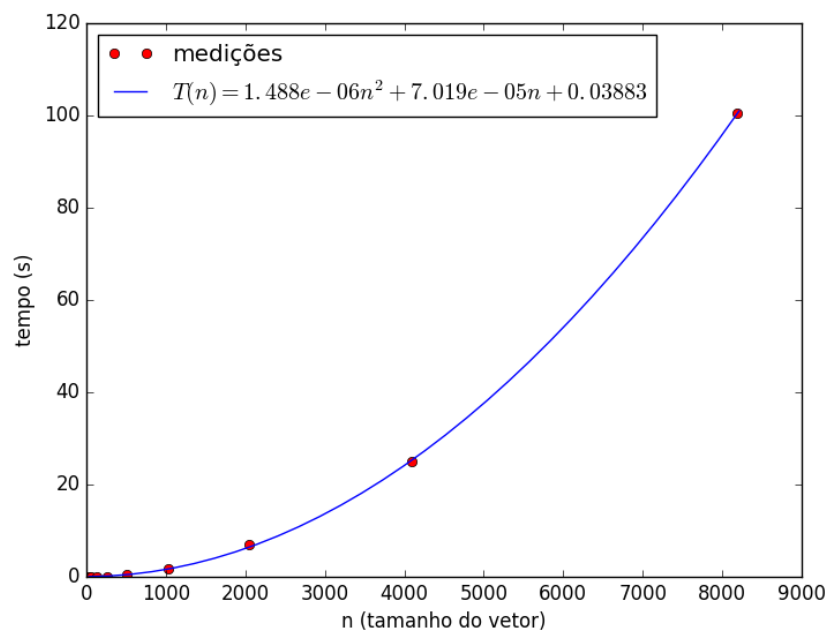


Figura 2.21: Bolha com relação ao tempo com vetor quase decrescente 30%.

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

$$1.488 * 10^{-6} * n^2 + 7.019 * 10^{-5} * n + 0.03883 = 2.744875548e + 13$$

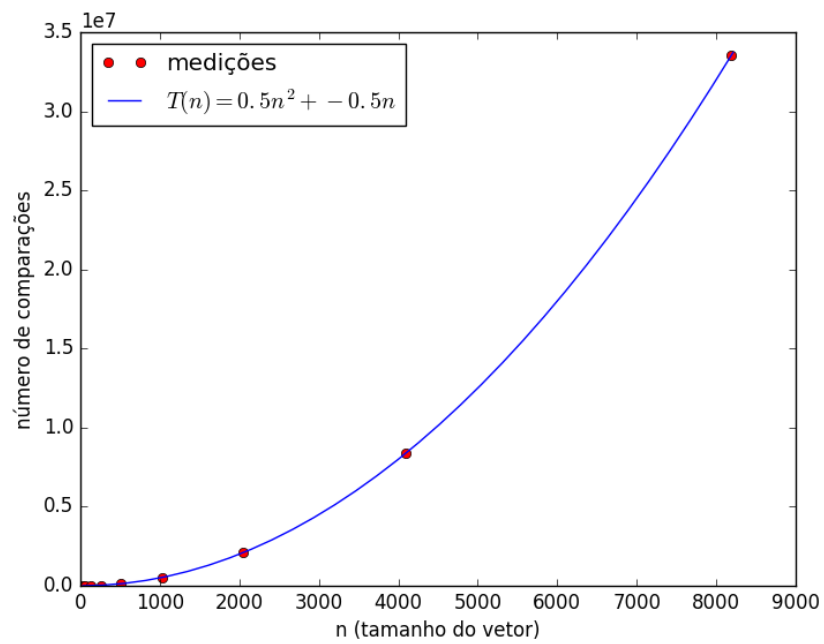


Figura 2.22: Bolha com relação ao número de comparações com vetor quase decrescente 30%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001454
64	2016	0.006385
128	8128	0.026360
256	32640	0.095272
512	130816	0.395558
1024	523776	1.689550
2048	2096128	5.883790
4096	8386560	29.291700
8192	33550336	111.649000

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

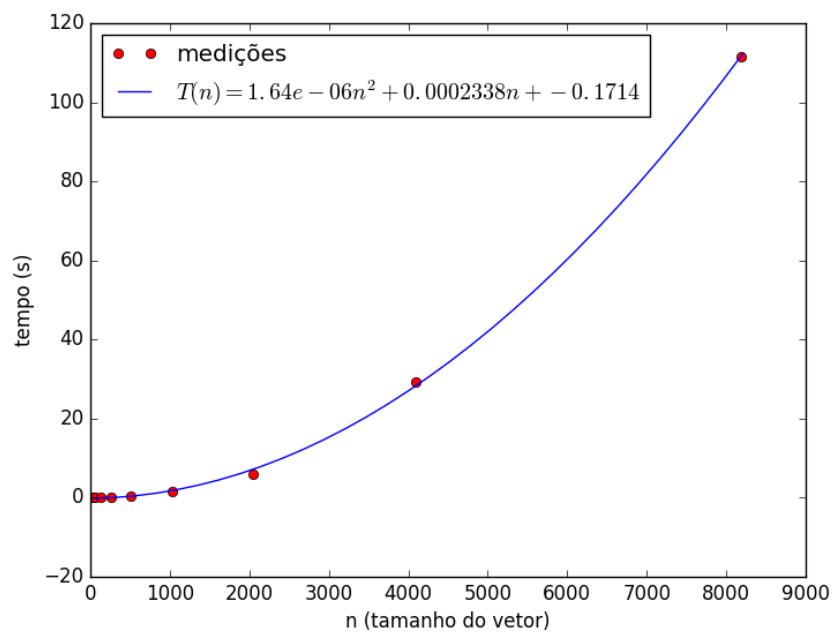


Figura 2.23: Bolha com relação ao tempo com vetor quase decrescente 40%.

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

$$1.64 * 10^{-6} * n^2 + 0.0002338 * n - 0.1714 = 3.025266129e + 13$$

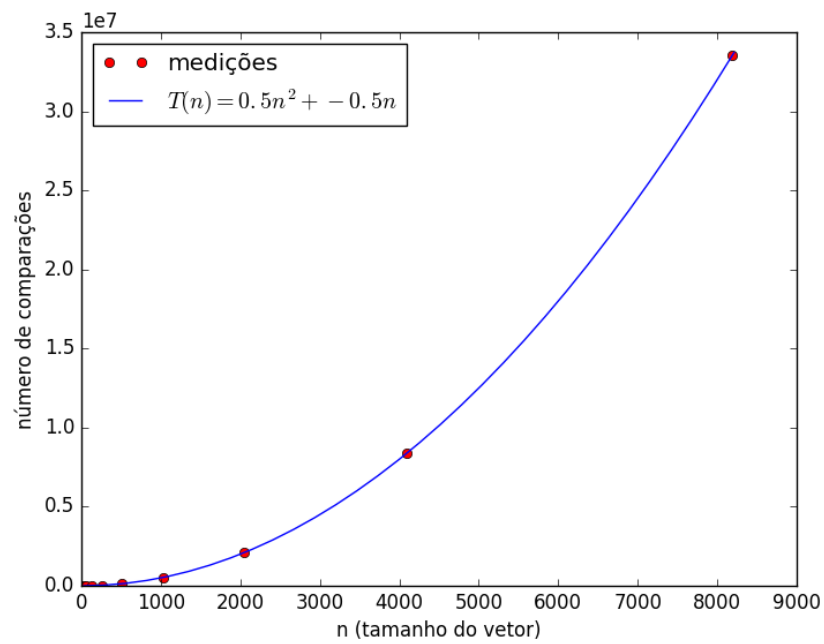


Figura 2.24: Bolha com relação ao número de comparações com vetor quase decrescente 40%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

n	comparações	tempo(s)
32	496	0.001486
64	2016	0.006156
128	8128	0.024230
256	32640	0.097495
512	130816	0.375425
1024	523776	1.424260
2048	2096128	5.997710
4096	8386560	23.341900
8192	33550336	98.741700

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

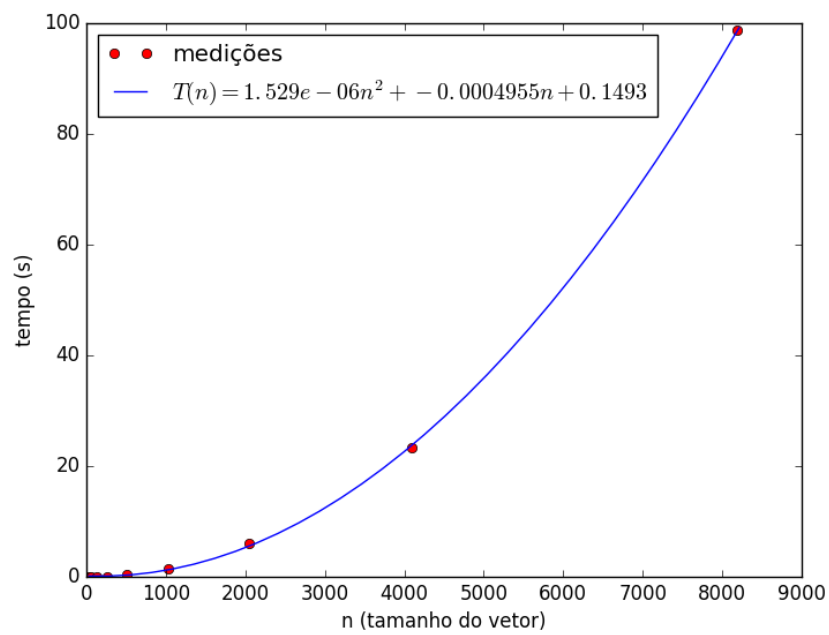


Figura 2.25: Bolha com relação ao tempo com vetor quase decrescente 50%.

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

$$1.529 * 10^{-6} * n^2 + 0.0004955 * n + 0.1493 = 2.820507382e + 13$$

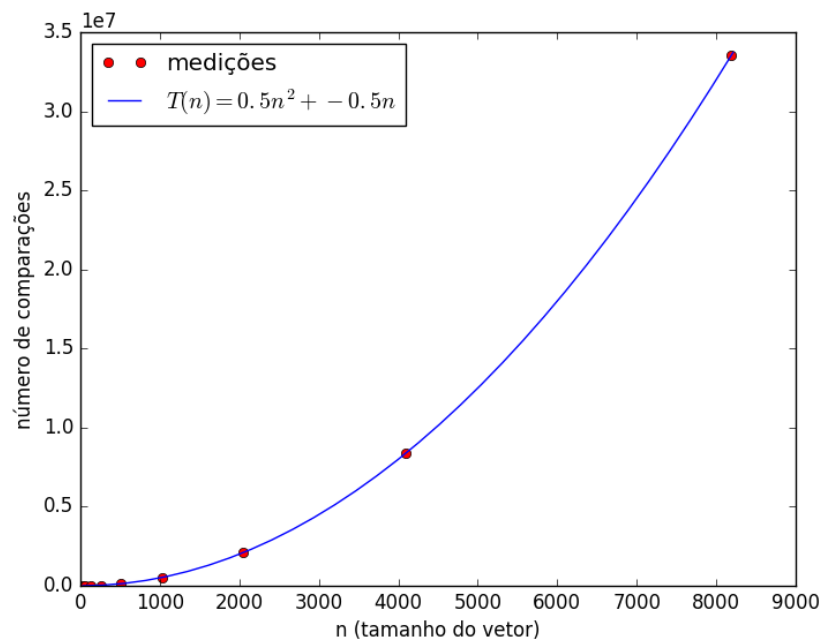


Figura 2.26: Bolha com relação ao número de comparações com vetor quase decrescente 50%.

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

$$0.5 * n^2 - 0.5 * n = 9.223372035e + 18$$

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)
```
