

Análise experimental de algoritmos usando Python

Patrícia Mariana Ramos Marcolino

`pmmarcolino@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 countingsort	8
2.2	A análise do grafico para 2^{32} segue abaixo para countingsort	9
2.3	A análise do grafico para 2^{32} segue abaixo para countingsort	10
2.4	A análise do grafico para 2^{32} segue abaixo para countingsort	11
2.5	A análise do grafico para 2^{32} segue abaixo para countingsort	12
2.6	A análise do grafico para 2^{32} segue abaixo para countingsort	13
2.7	A análise do grafico para 2^{32} segue abaixo para countingsort	14
2.8	A análise do grafico para 2^{32} segue abaixo para countingsort	15
2.9	A análise do grafico para 2^{32} segue abaixo para countingsort	16
2.10	A análise do grafico para 2^{32} segue abaixo para countingsort	17
2.11	A análise do grafico para 2^{32} segue abaixo para countingsort	19
2.12	A análise do grafico para 2^{32} segue abaixo para countingsort	20
2.13	A análise do grafico para 2^{32} segue abaixo para countingsort	21
2.14	A análise do grafico para 2^{32} segue abaixo para countingsort	22
2.15	A análise do grafico para 2^{32} segue abaixo para countingsort	23
2.16	A análise do grafico para 2^{32} segue abaixo para countingsort	24
2.17	A análise do grafico para 2^{32} segue abaixo para countingsort	25
2.18	A análise do grafico para 2^{32} segue abaixo para countingsort	26
2.19	A análise do grafico para 2^{32} segue abaixo para countingsort	27
2.20	A análise do grafico para 2^{32} segue abaixo para countingsort	28
2.21	A análise do grafico para 2^{32} segue abaixo para countingsort	30
2.22	A análise do grafico para 2^{32} segue abaixo para countingsort	31
2.23	A análise do grafico para 2^{32} segue abaixo para countingsort	32
2.24	A análise do grafico para 2^{32} segue abaixo para countingsort	33
2.25	A análise do grafico para 2^{32} segue abaixo para countingsort	34
2.26	A análise do grafico para 2^{32} segue abaixo para countingsort	35

Lista de Tabelas

2.1	Tabela com vetor teste aleatório: A linha te interesse analisada para este caso é a 13.	7
2.2	Tabela com vetor teste crescente: A linha te interesse analisada para este caso é a 13.	7
2.3	Tabela com vetor teste decrescente: A linha te interesse analisada para este caso é a 13.	8
2.4	Tabela com vetor teste quase crescente 10%: A linha te interesse analisada para este caso é a 13.	9
2.5	Tabela com vetor teste quase crescente 20%: A linha te interesse analisada para este caso é a 13.	10
2.6	Tabela com vetor teste quase crescente 30%: A linha te interesse analisada para este caso é a 13.	18
2.7	Tabela com vetor teste quase crescente 40%: A linha te interesse analisada para este caso é a 13.	18
2.8	Tabela com vetor teste quase crescente 50%: A linha te interesse analisada para este caso é a 13.	19
2.9	Tabela com vetor teste quase decrescente 10%: A linha te interesse analisada para este caso é a 13.	20
2.10	Tabela com vetor teste quase decrescente 20%: A linha te interesse analisada para este caso é a 13.	21
2.11	Tabela com vetor teste quase decrescente 30%: A linha te interesse analisada para este caso é a 13.	29
2.12	Tabela com vetor teste quase decrescente 40%: A linha te interesse analisada para este caso é a 13.	29
2.13	Tabela com vetor teste quase decrescente 50%: A linha te interesse analisada para este caso é a 13.	30

Lista de Listagens

A.1	../countingsort/countingsort.py	36
B.1	../countingsort/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 ../countingsort/countingsort.py	36
B Arquivo ../countingsort/ensaio.py	37

Capítulo 1

Análise

O algoritmo não faz comparações entre elementos de A . Sua complexidade deve ser medida com base nas outras operações (aritméticas, atribuições, etc.) Claramente, o número de tais operações é uma função em $O(n+k)$, já que temos dois loops simples com n iterações e dois com k iterações. Assim, quando $k \in O(n)$, este algoritmo tem complexidade $O(n)$.

Capítulo 2

Resultados

2.1 Tabelas

n	comparações	tempo(s)
32	33	0.000976
64	65	0.001012
128	129	0.001295
256	257	0.001650
512	513	0.002271
1024	1025	0.003947
2048	2049	0.006848
4096	4097	0.012388
8192	8193	0.025360

Tabela 2.1: Tabela com vetor teste aleatório: A linha de interesse analisada para este caso é a 13.

n	comparações	tempo(s)
32	33	0.000893
64	65	0.001035
128	129	0.001240
256	257	0.001694
512	513	0.002266
1024	1025	0.004139
2048	2049	0.006838
4096	4097	0.013341
8192	8193	0.025272

Tabela 2.2: Tabela com vetor teste crescente: A linha de interesse analisada para este caso é a 13.

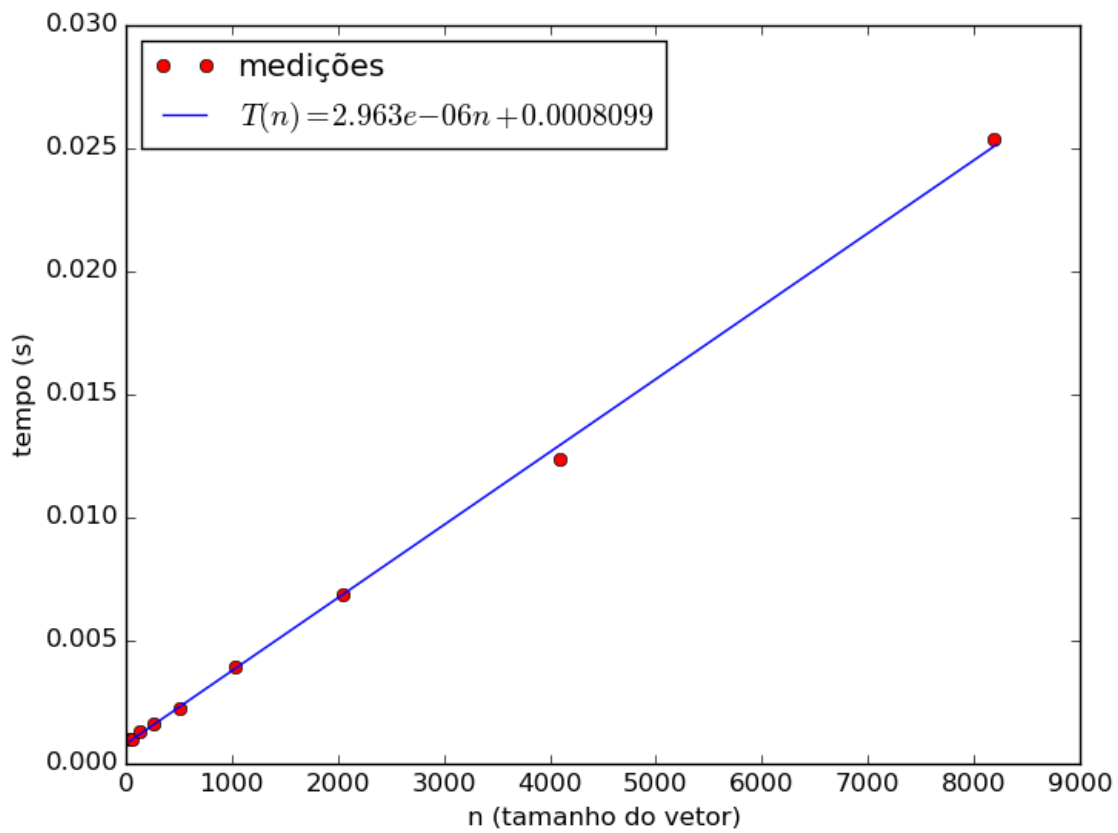


Figura 2.1: A análise do grafico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = 2.963e - 6 * n - 0.0008099$ e para o $n = 2^{32}$, $T(2^{32}) = 12725.987288148$

n	comparações	tempo(s)
32	33	0.000894
64	65	0.001013
128	129	0.001249
256	257	0.001672
512	513	0.002324
1024	1025	0.003906
2048	2049	0.007068
4096	4097	0.012608
8192	8193	0.025750

Tabela 2.3: Tabela com vetor teste decrescente: A linha de interesse analisada para este caso é a 13.

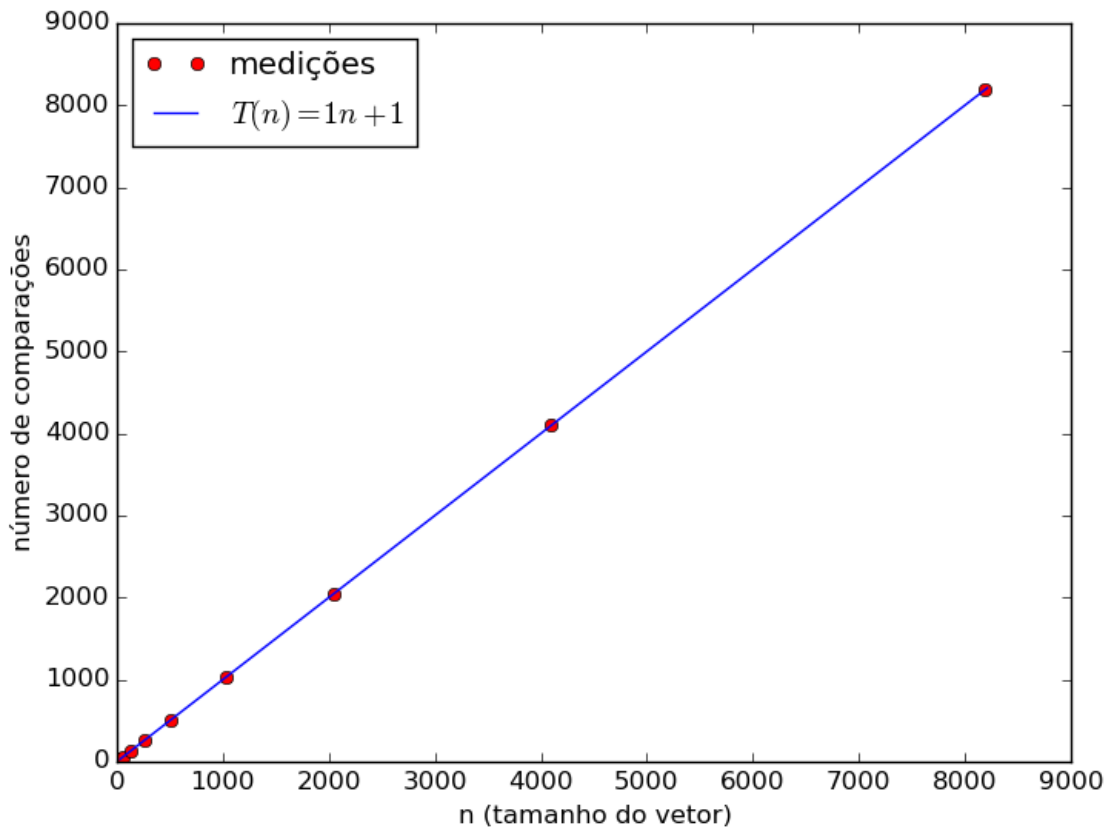


Figura 2.2: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

n	comparações	tempo(s)
32	33	0.000959
64	65	0.001051
128	129	0.001267
256	257	0.001669
512	513	0.002349
1024	1025	0.003928
2048	2049	0.006814
4096	4097	0.012612
8192	8193	0.024086

Tabela 2.4: Tabela com vetor teste quase crescente 10%. A linha de interesse analisada para este caso é a 13.

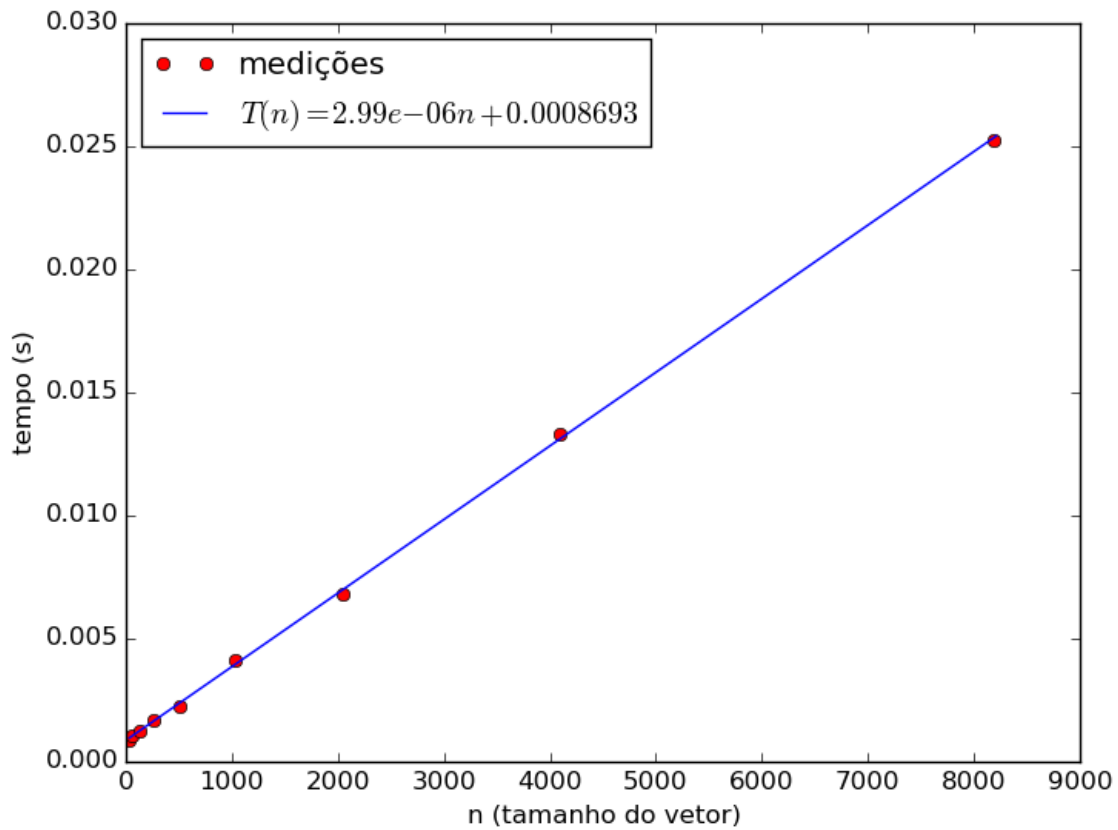


Figura 2.3: A análise do grafico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 2.99e - 6 * n - 0.0008693$ e para o $n = 2^{32}$,
 $T(2^{32}) = 12841.95134574$

n	comparações	tempo(s)
32	33	0.000958
64	65	0.001027
128	129	0.001192
256	257	0.001584
512	513	0.002341
1024	1025	0.004560
2048	2049	0.007211
4096	4097	0.016147
8192	8193	0.024661

Tabela 2.5: Tabela com vetor teste quase crescente 20%: A linha te interesse analisada para este caso é a 13.

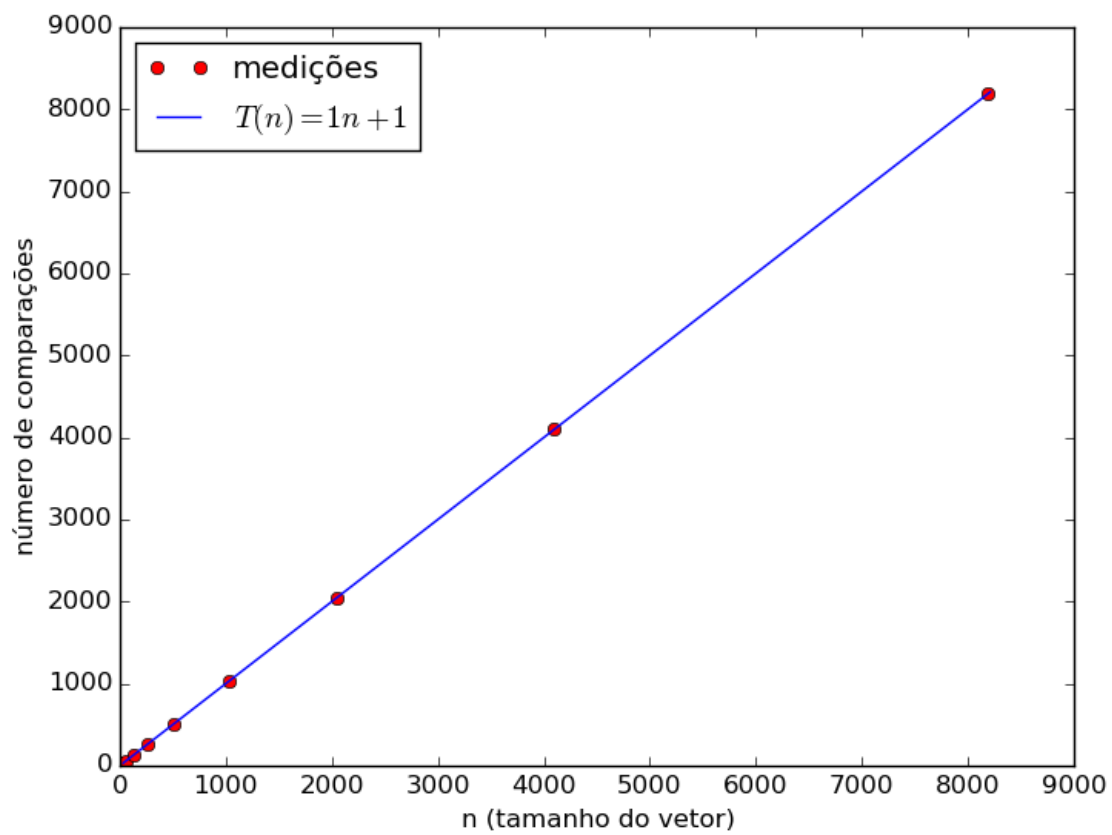


Figura 2.4: A análise do gráfico para 2^{32} segue abaixo para counting sort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

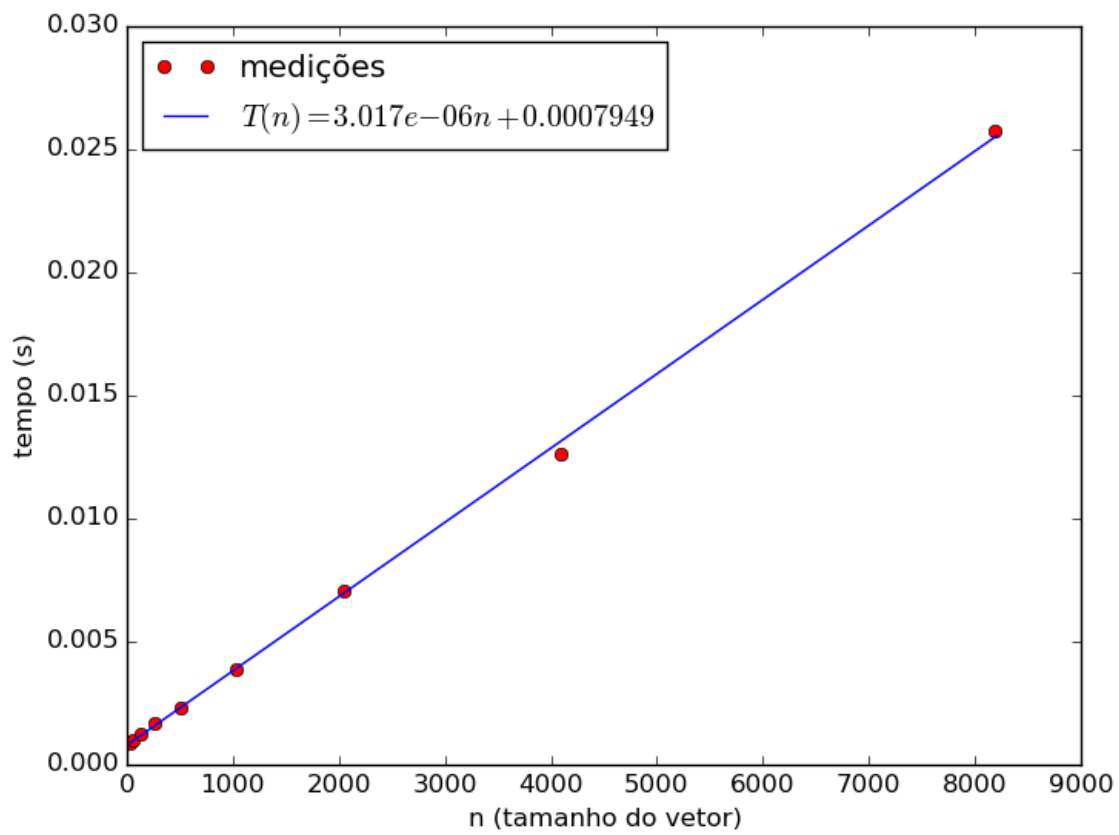


Figura 2.5: A análise do grafico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 3.017e - 6 * n - 0.0007949$ e para o $n = 2^{32}$,
 $T(2^{32}) = 12957.915537132$

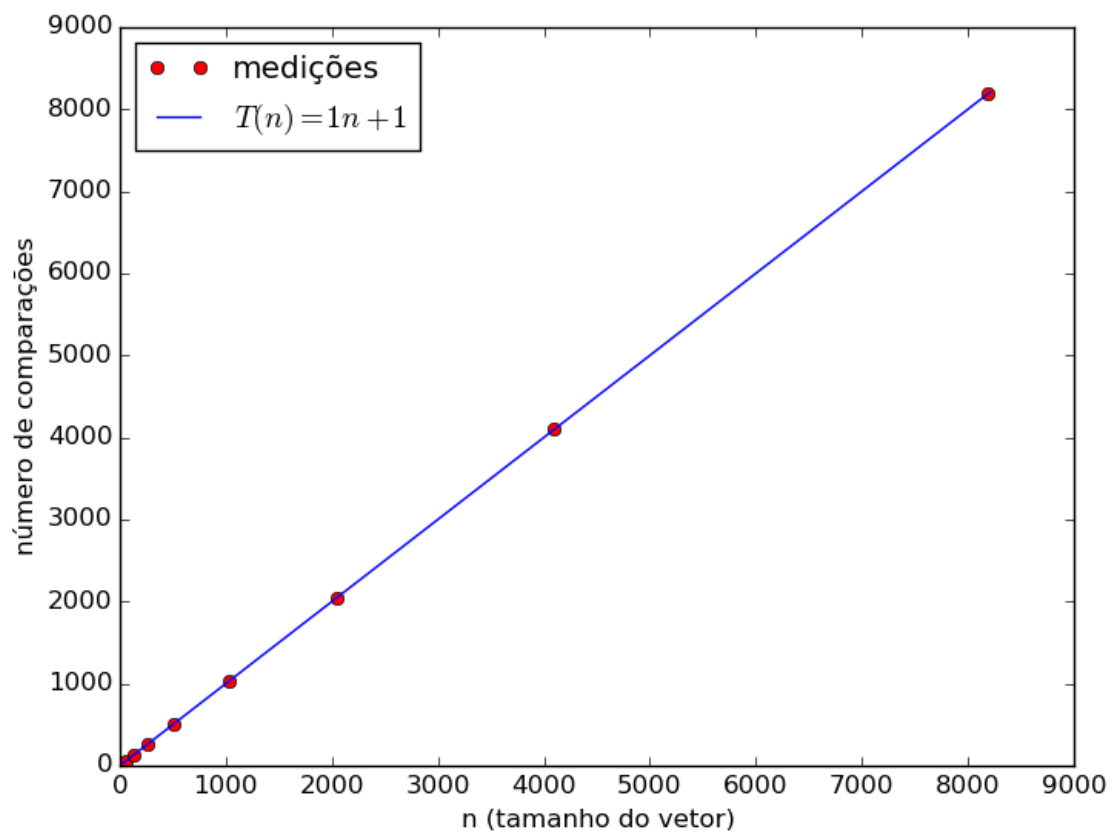


Figura 2.6: A análise do gráfico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$

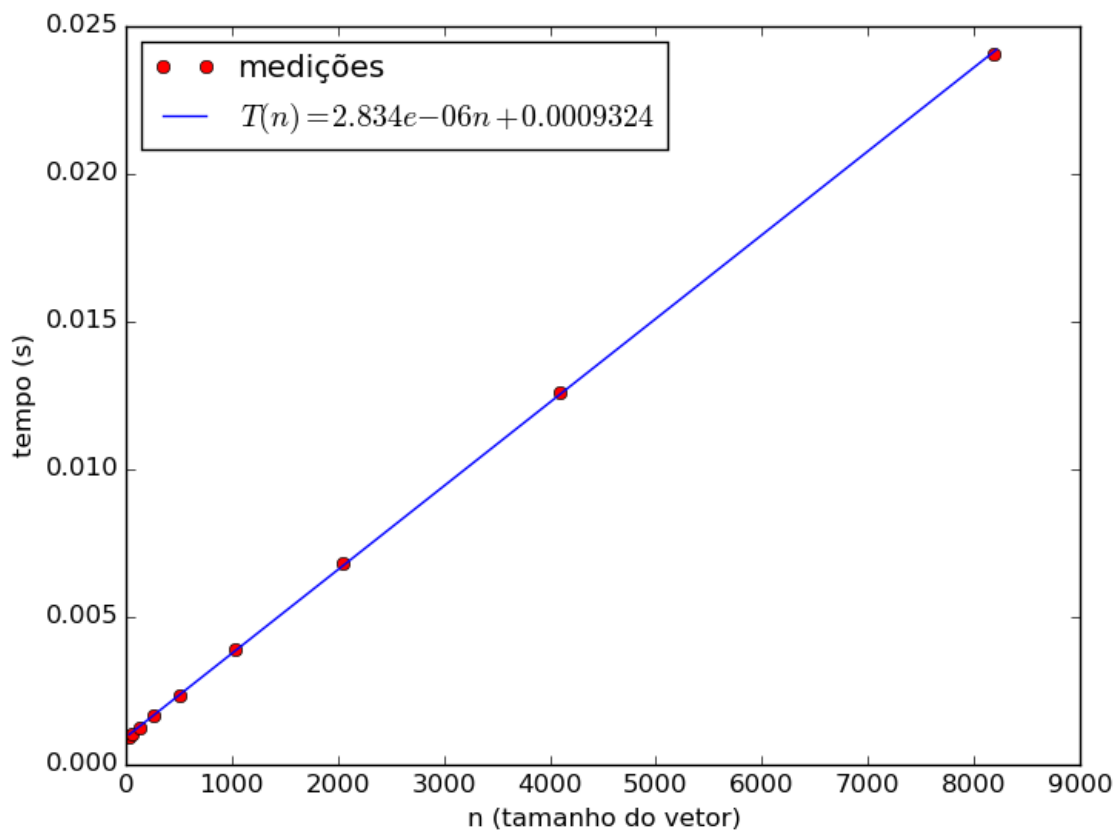


Figura 2.7: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = 2.834e - 6 * n - 0.0009324$ e para o $n = 2^{32}$,
 $T(2^{32}) = 12171.936384464$

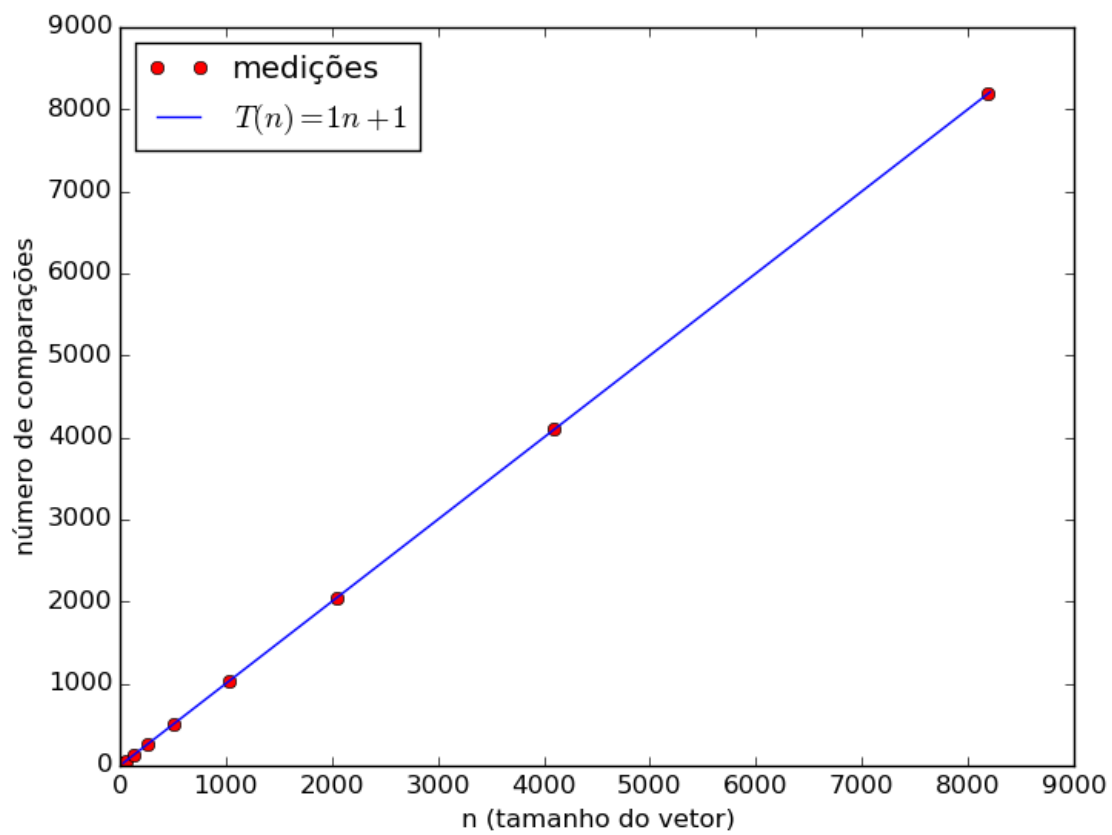


Figura 2.8: A análise do gráfico para 2^{32} segue abaixo para counting sort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

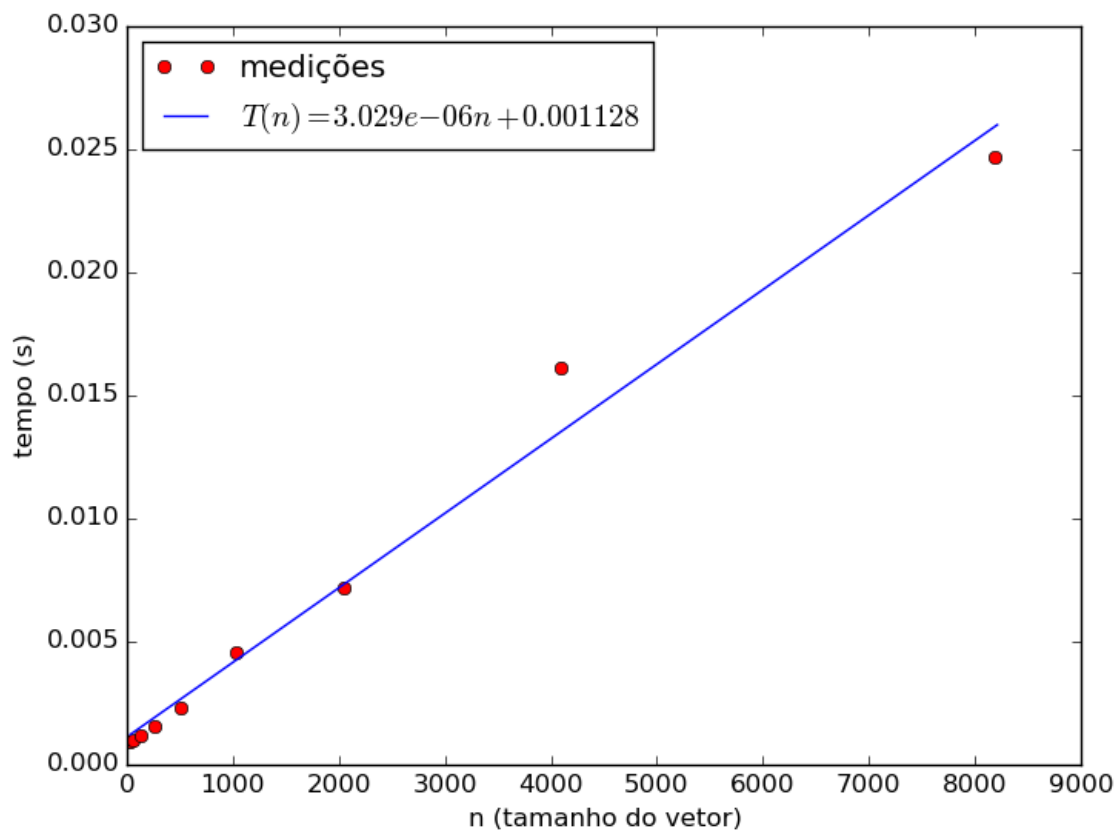


Figura 2.9: A análise do grafico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 3.029e - 6 * n - 0.0001128$ e para o $n = 2^{32}$,
 $T(2^{32}) = 13009.455826784$

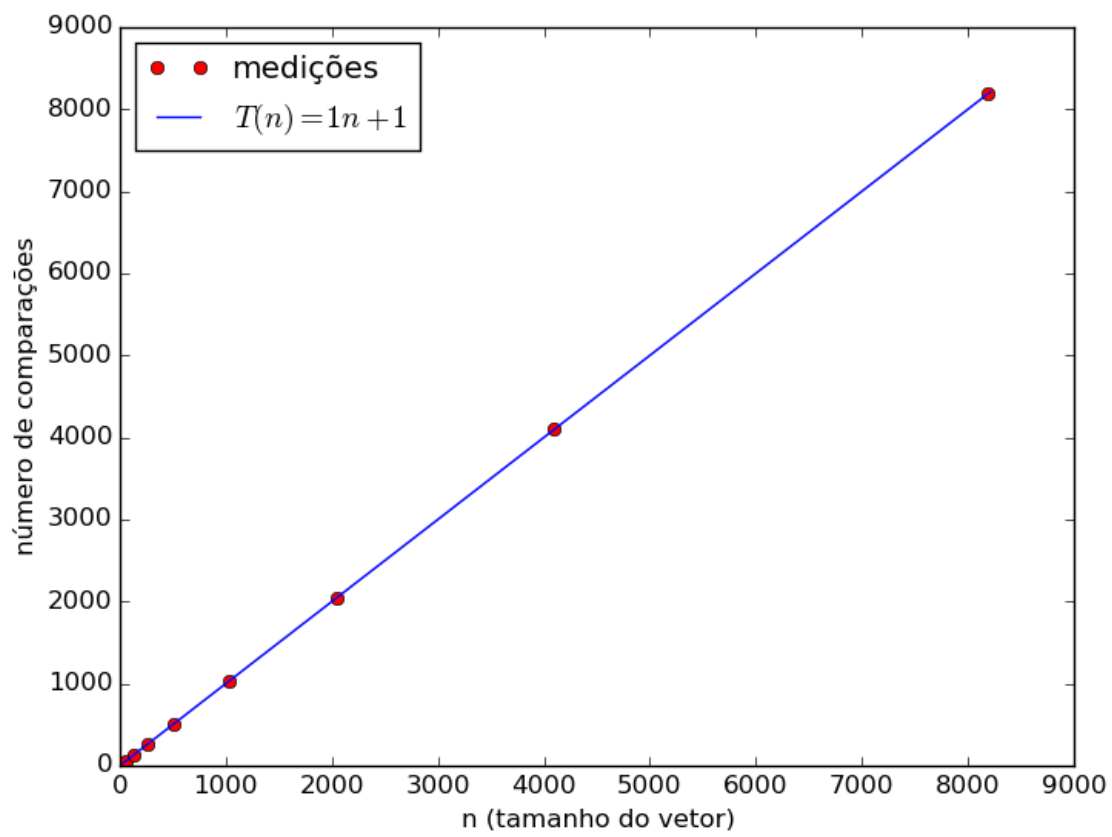


Figura 2.10: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

n	comparações	tempo(s)
32	33	0.000902
64	65	0.001071
128	129	0.001190
256	257	0.001560
512	513	0.002343
1024	1025	0.004084
2048	2049	0.006719
4096	4097	0.013096
8192	8193	0.024099

Tabela 2.6: Tabela com vetor teste quase crescente 30%: A linha de interesse analisada para este caso é a 13.

n	comparações	tempo(s)
32	33	0.000946
64	65	0.000999
128	129	0.001233
256	257	0.001511
512	513	0.002411
1024	1025	0.004073
2048	2049	0.006999
4096	4097	0.012641
8192	8193	0.024806

Tabela 2.7: Tabela com vetor teste quase crescente 40%: A linha de interesse analisada para este caso é a 13.

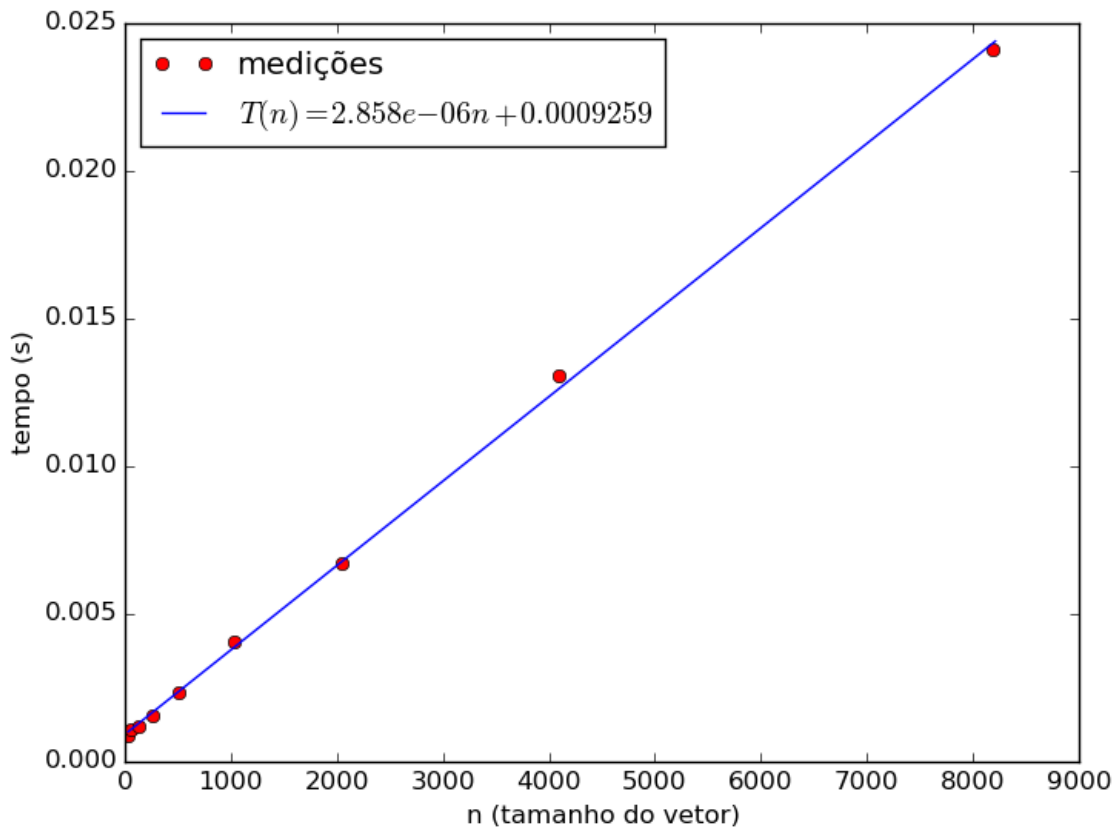


Figura 2.11: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = 2.858e - 6 * n - 0.0009259$ e para o $n = 2^{32}$, $T(2^{32}) = 12275.015606068$

n	comparações	tempo(s)
32	33	0.000923
64	65	0.001042
128	129	0.001211
256	257	0.001560
512	513	0.002591
1024	1025	0.004088
2048	2049	0.007676
4096	4097	0.013312
8192	8193	0.026769

Tabela 2.8: Tabela com vetor teste quase crescente 50%: A linha de interesse analisada para este caso é a 13.

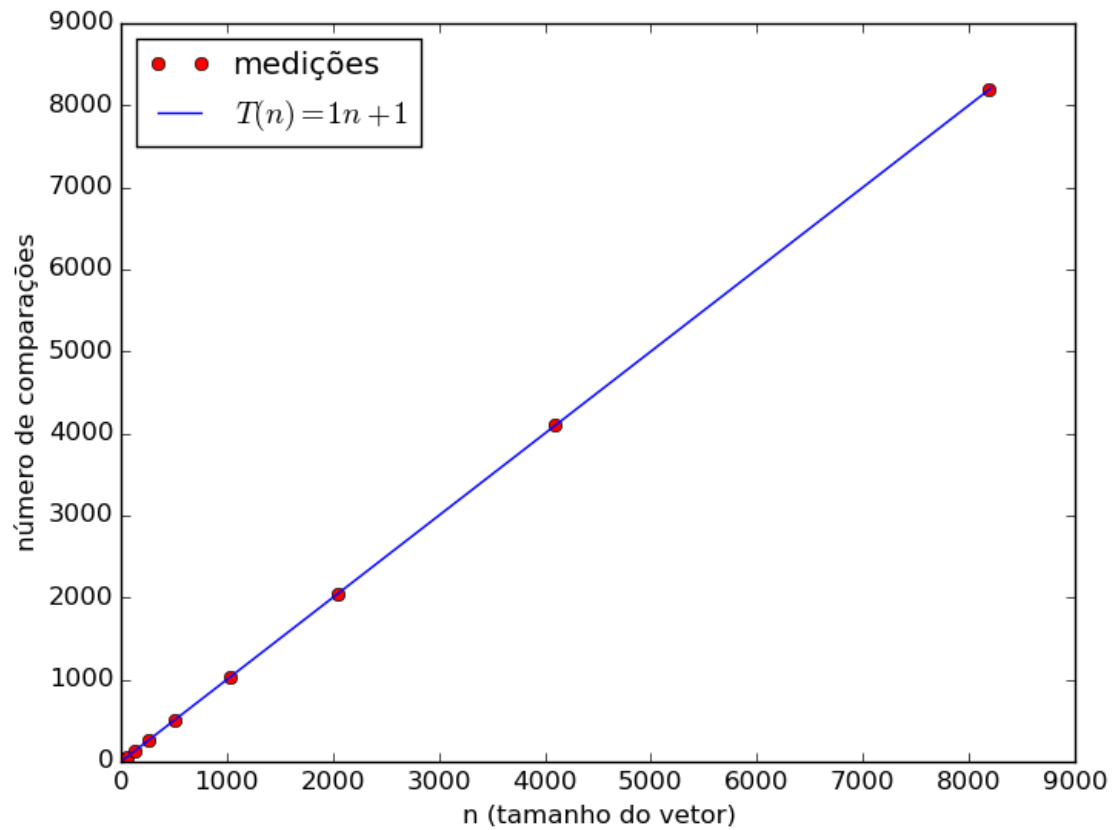


Figura 2.12: A análise do gráfico para 2^{32} segue abaixo para counting sort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

n	comparações	tempo(s)
32	33	0.000910
64	65	0.001056
128	129	0.001275
256	257	0.001599
512	513	0.002361
1024	1025	0.003694
2048	2049	0.006854
4096	4097	0.013677
8192	8193	0.026185

Tabela 2.9: Tabela com vetor teste quase decrescente 10%. A linha de interesse analisada para este caso é a 13.

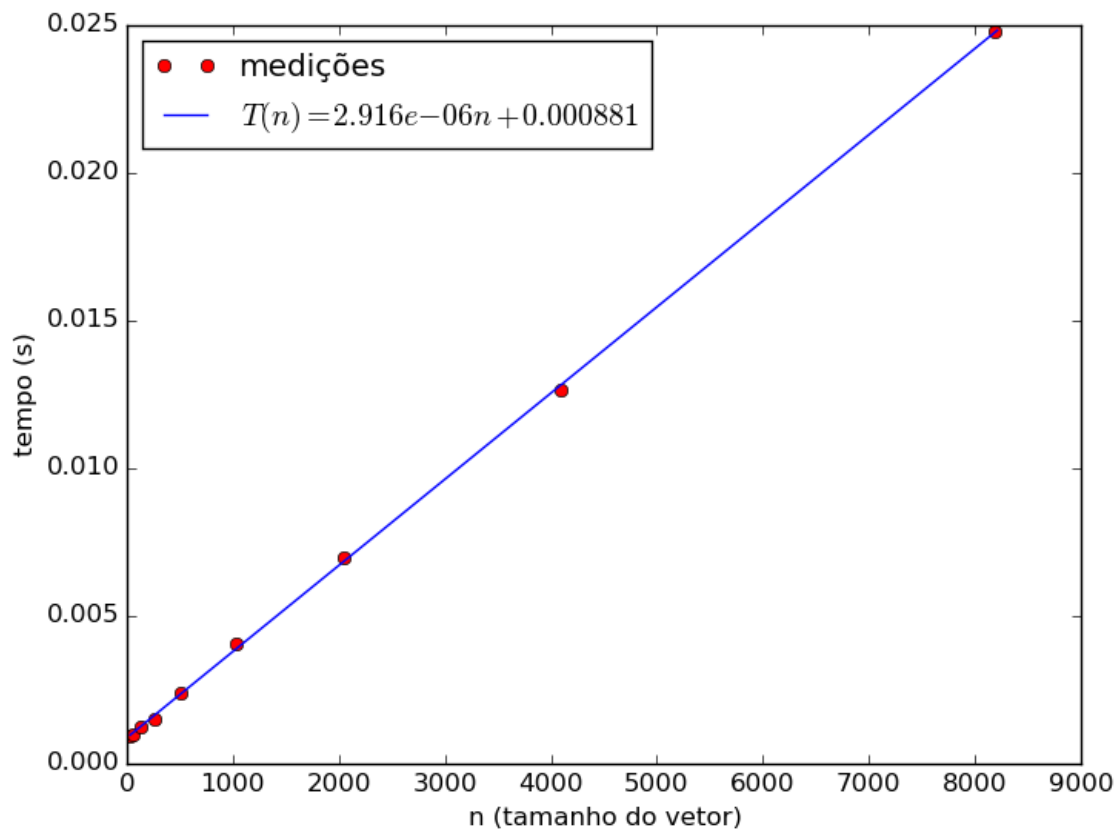


Figura 2.13: A análise do gráfico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 2.916e - 6 * n - 0.000881$ e para o $n = 2^{32}$,
 $T(2^{32}) = 12524.123754136$

n	comparações	tempo(s)
32	33	0.000944
64	65	0.001062
128	129	0.001279
256	257	0.001567
512	513	0.002408
1024	1025	0.003870
2048	2049	0.006902
4096	4097	0.012783
8192	8193	0.024632

Tabela 2.10: Tabela com vetor teste quase decrescente 20%: A linha de interesse analisada para este caso é a 13.

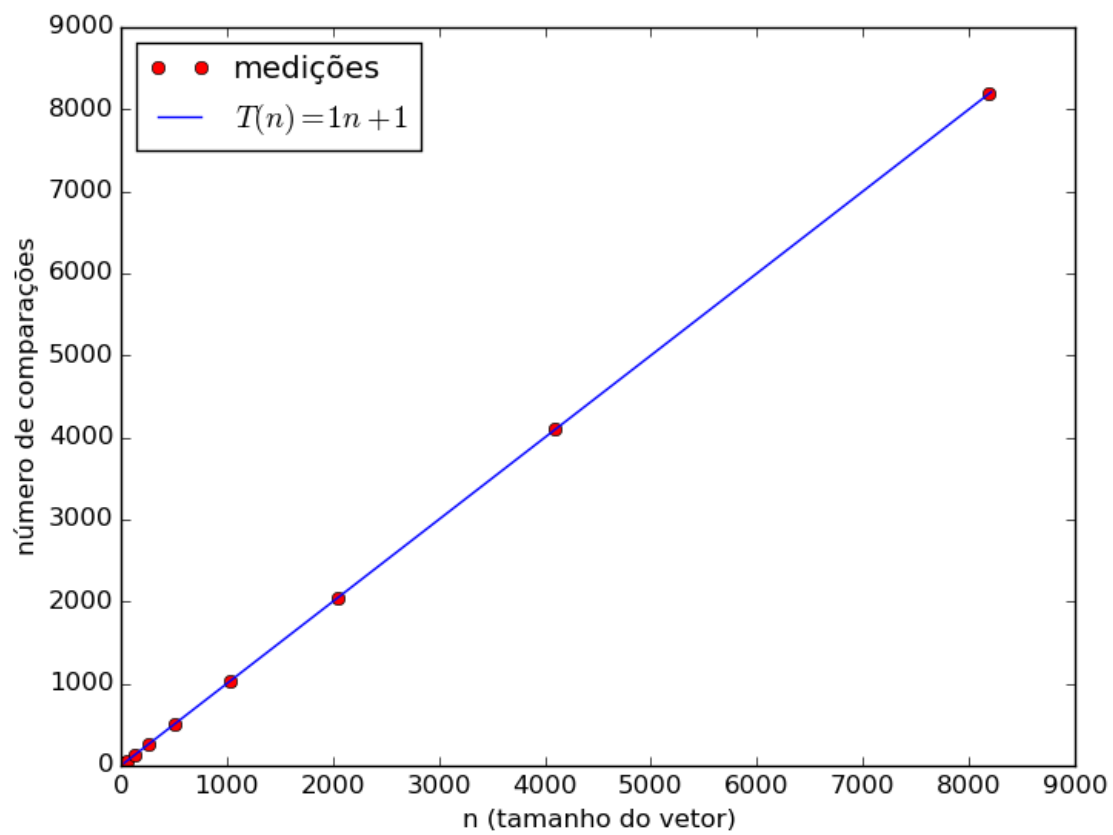


Figura 2.14: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

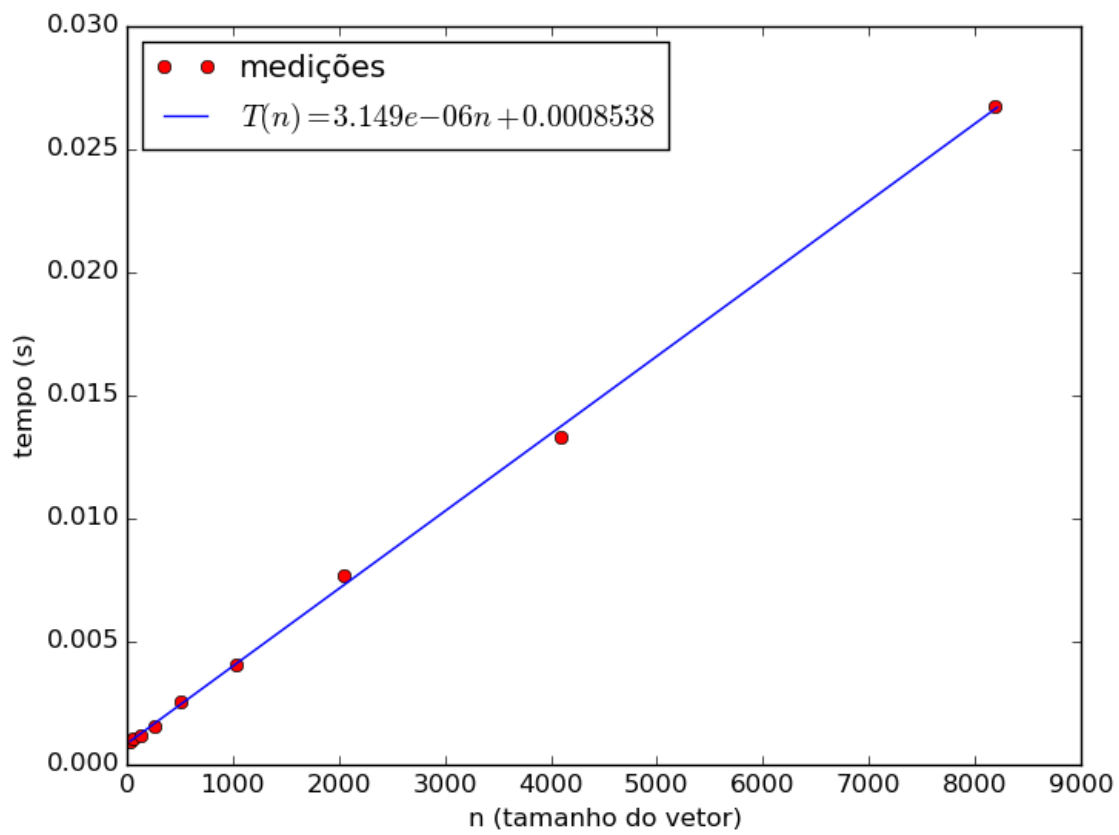


Figura 2.15: A análise do gráfico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 3.149e - 6 * n - 0.0008538$ e para o $n = 2^{32}$,
 $T(2^{32}) = 13524.851161304$

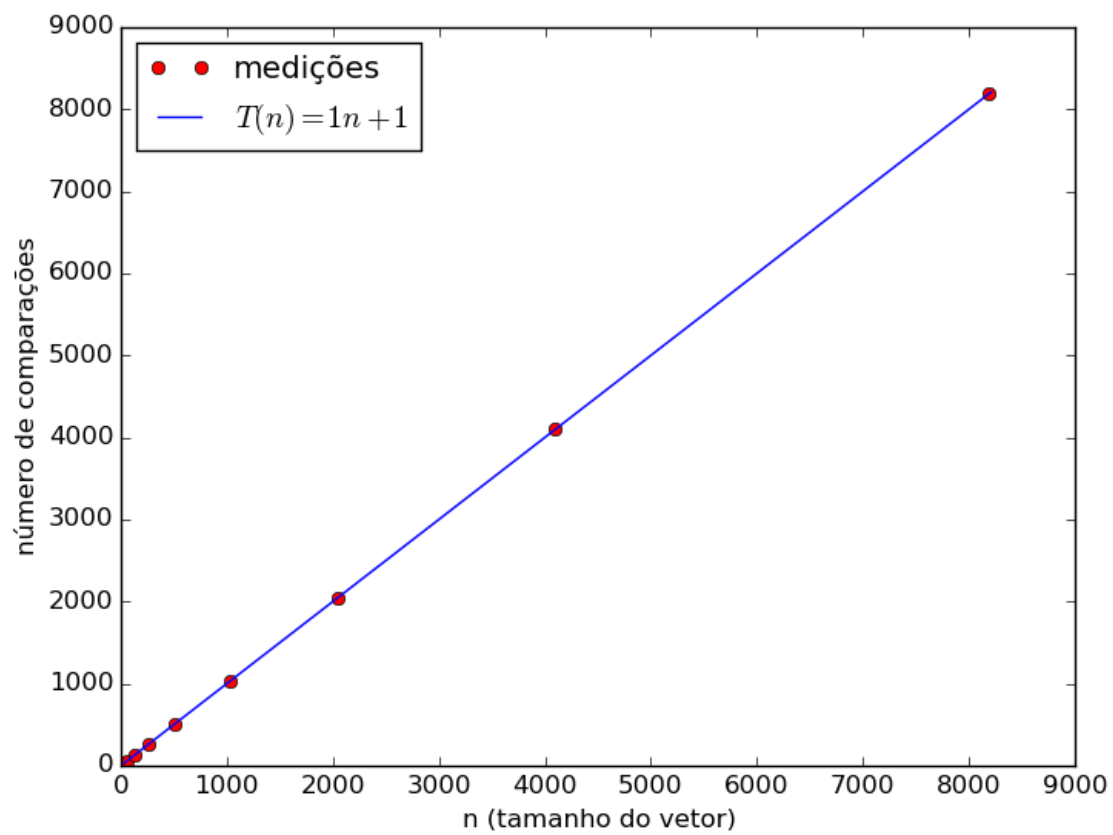


Figura 2.16: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

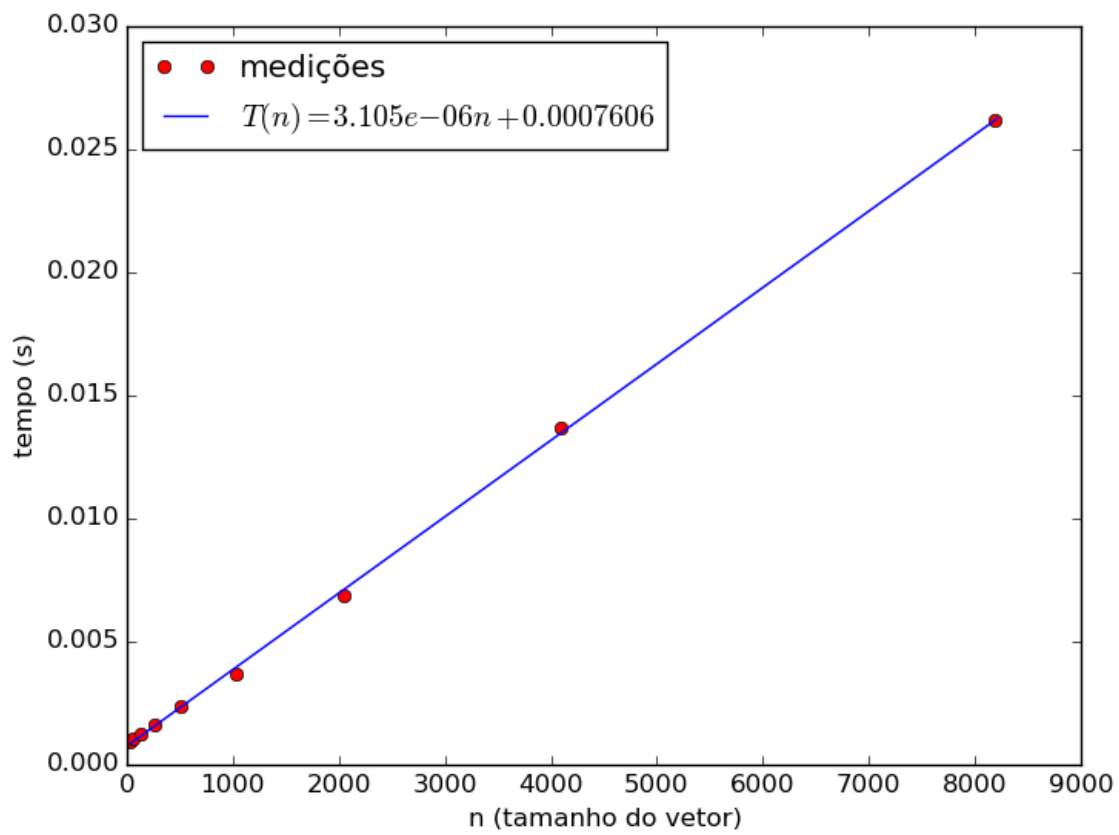


Figura 2.17: A análise do gráfico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 3.105e - 6 * n - 0.0007606$ e para o $n = 2^{32}$,
 $T(2^{32}) = 13335.87269348$

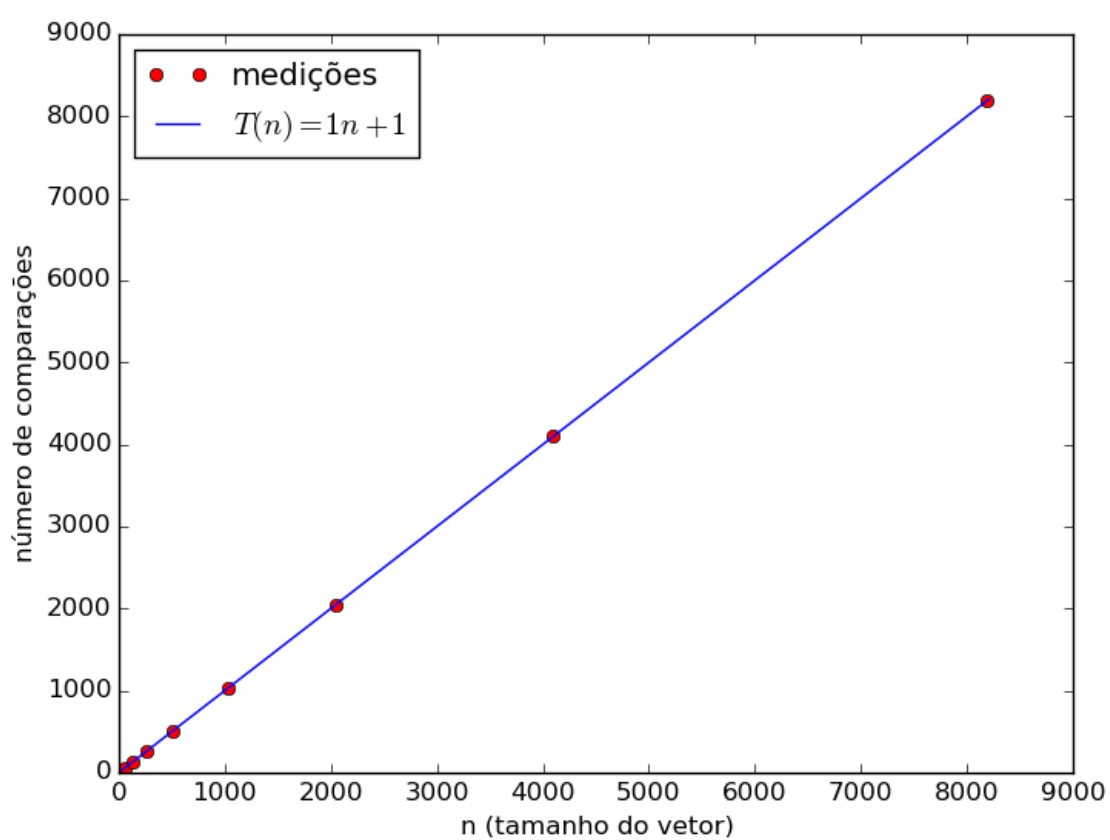


Figura 2.18: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

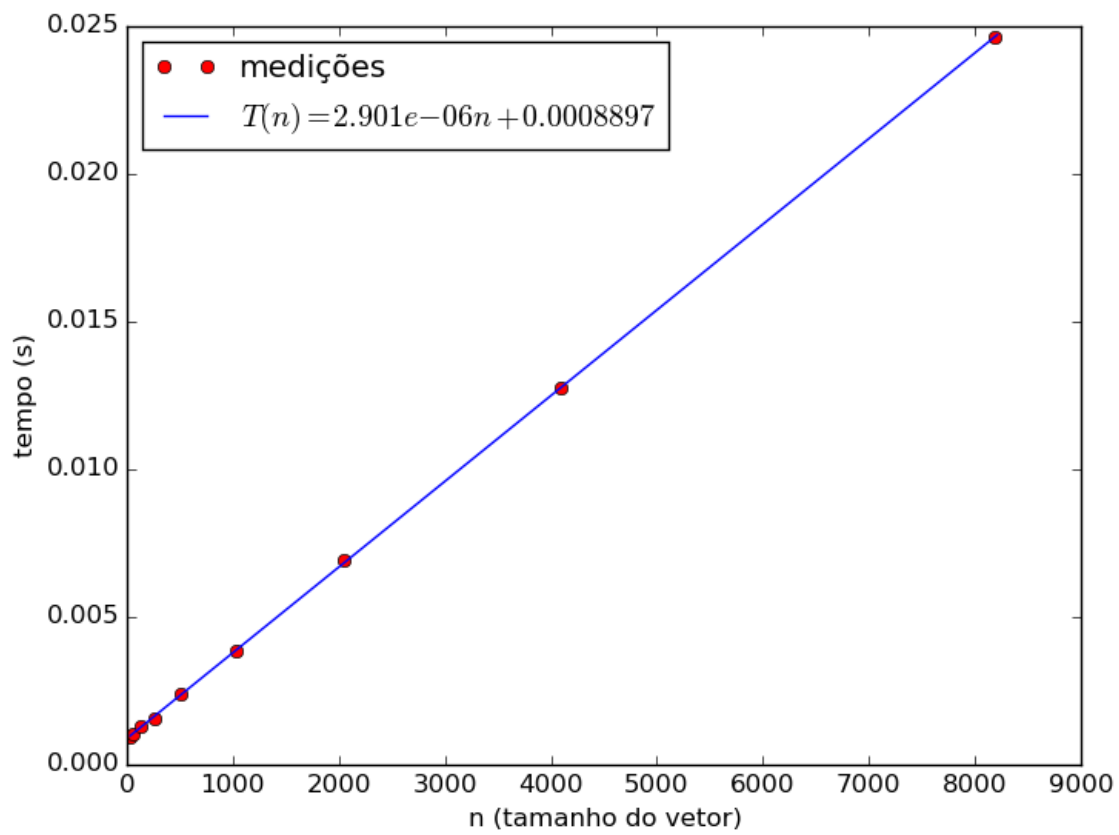


Figura 2.19: A análise do gráfico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = 2.901e - 6 * n - 0.0008897$ e para o $n = 2^{32}$,
 $T(2^{32}) = 12459.699235996$

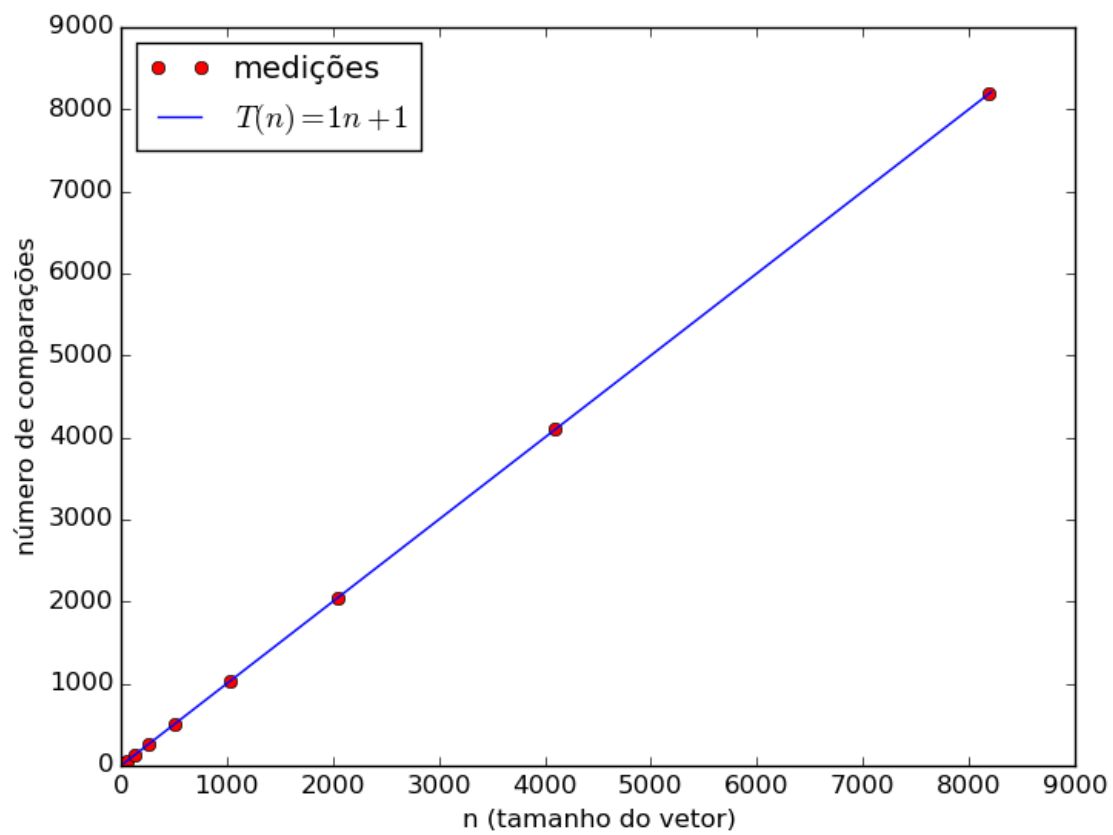


Figura 2.20: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

n	comparações	tempo(s)
32	33	0.000966
64	65	0.001029
128	129	0.001243
256	257	0.001740
512	513	0.002443
1024	1025	0.003878
2048	2049	0.006573
4096	4097	0.014319
8192	8193	0.025719

Tabela 2.11: Tabela com vetor teste quase decrescente 30%: A linha de interesse analisada para este caso é a 13.

n	comparações	tempo(s)
32	33	0.000952
64	65	0.001077
128	129	0.001266
256	257	0.001574
512	513	0.002256
1024	1025	0.004110
2048	2049	0.006739
4096	4097	0.012949
8192	8193	0.026137

Tabela 2.12: Tabela com vetor teste quase decrescente 40%: A linha de interesse analisada para este caso é a 13.

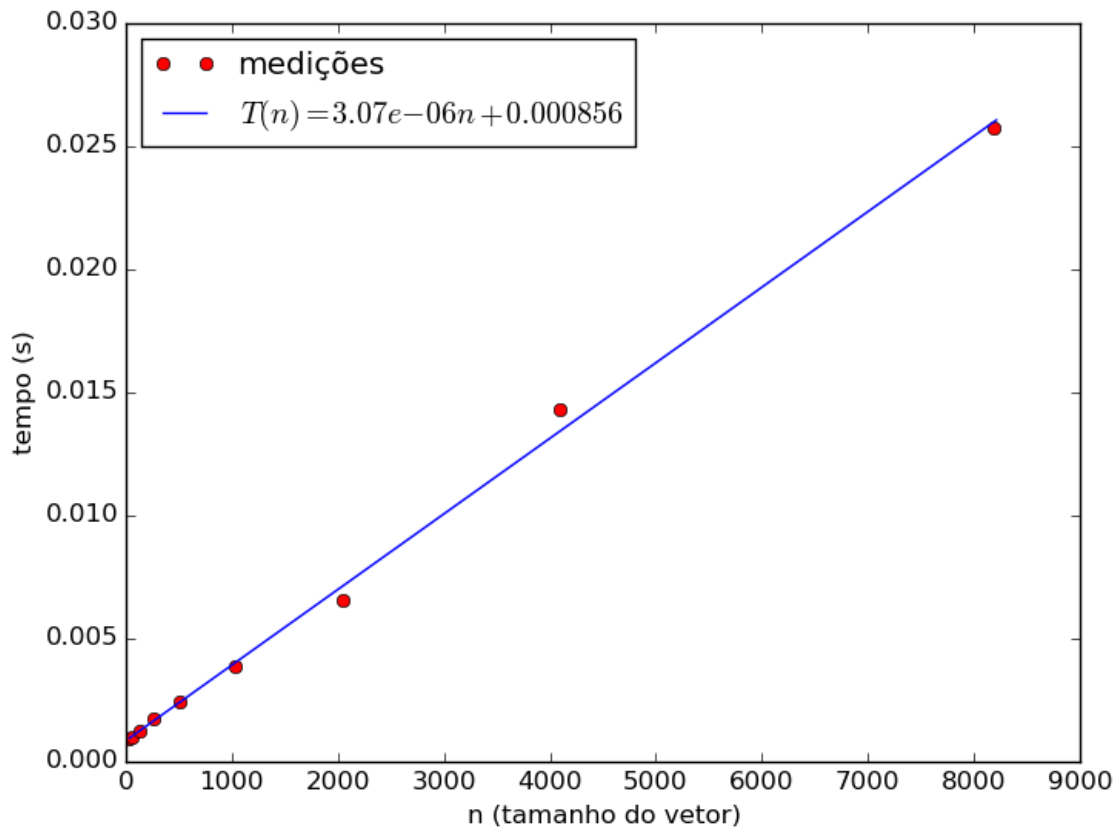


Figura 2.21: A análise do gráfico para 2^{32} segue abaixo para counting sort
Tendo a função $T(n) = 3.07e-6 * n - 0.000856$ e para o $n = 2^{32}$, $T(2^{32}) = 13185.54874272$

n	comparações	tempo(s)
32	33	0.000972
64	65	0.001001
128	129	0.001180
256	257	0.001592
512	513	0.002335
1024	1025	0.003802
2048	2049	0.006989
4096	4097	0.013460
8192	8193	0.026501

Tabela 2.13: Tabela com vetor teste quase decrescente 50%: A linha de interesse analisada para este caso é a 13.

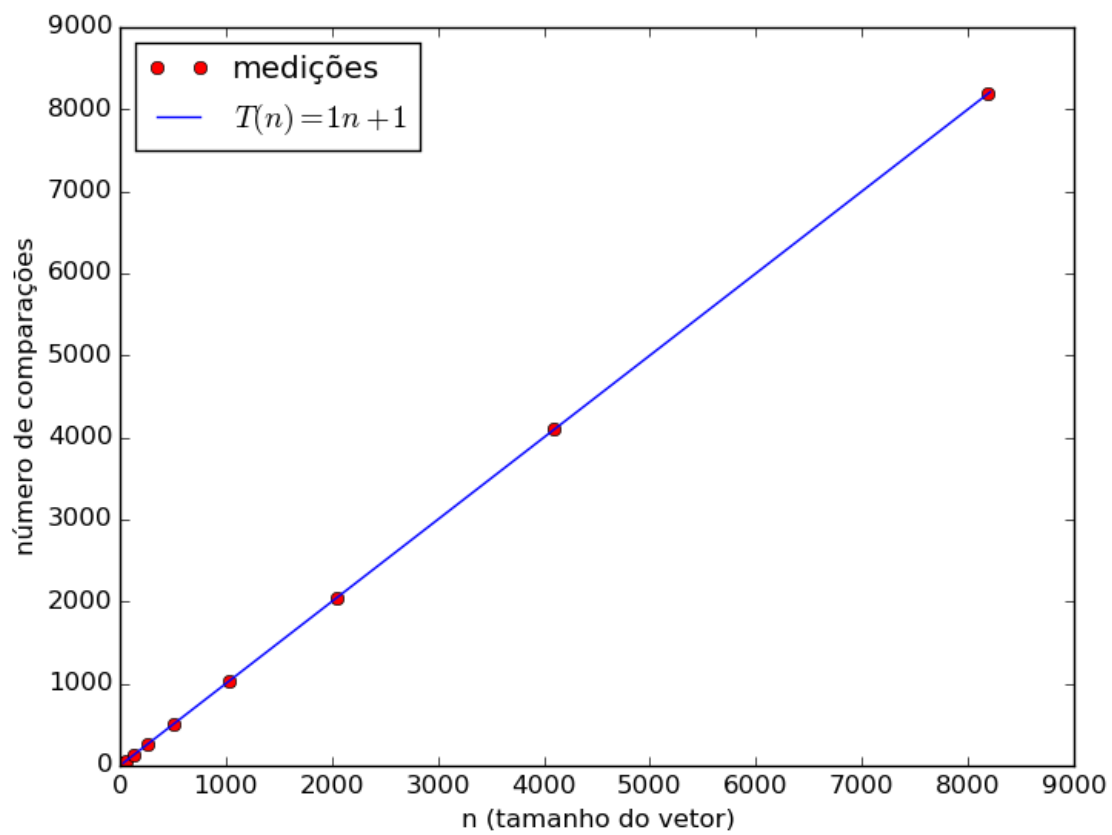


Figura 2.22: A análise do grafico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

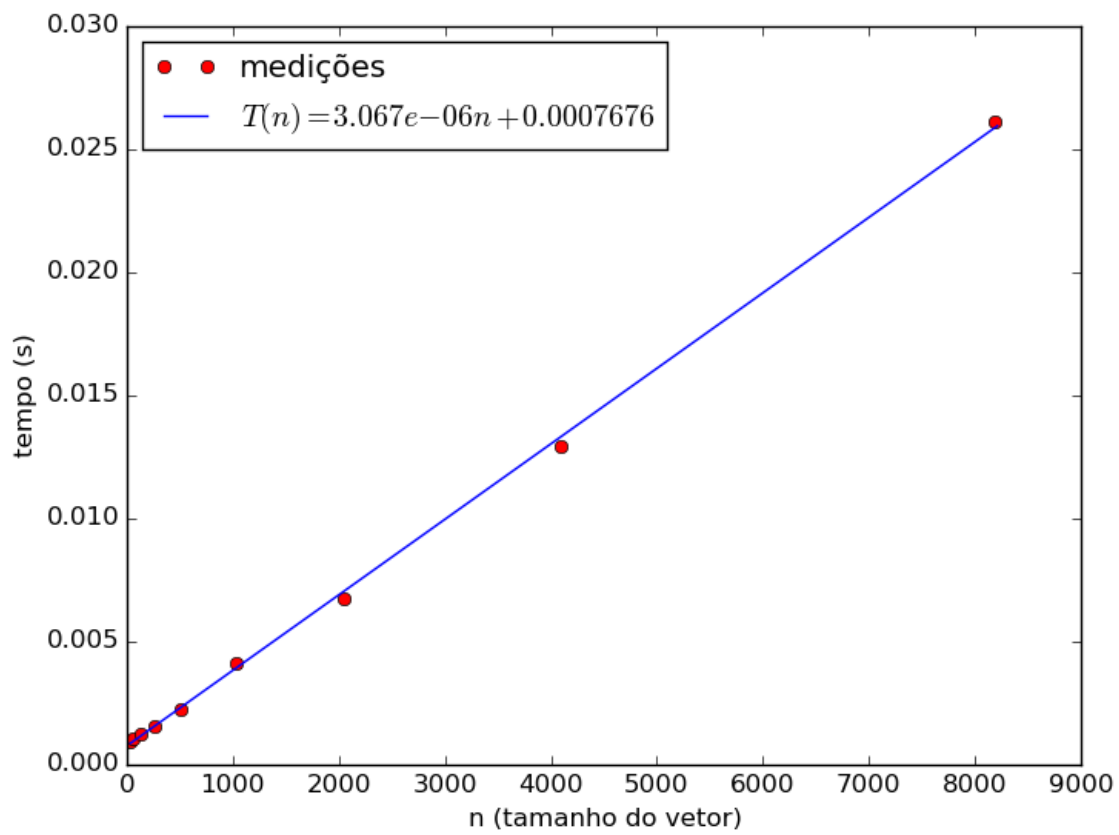


Figura 2.23: A análise do grafico para 2^{32} segue abaixo para counting sort
Tendo a função $T(n) = 3.067e - 6 * n - 0.0007676$ e para o $n = 2^{32}$,
 $T(2^{32}) = 13172.663929232$

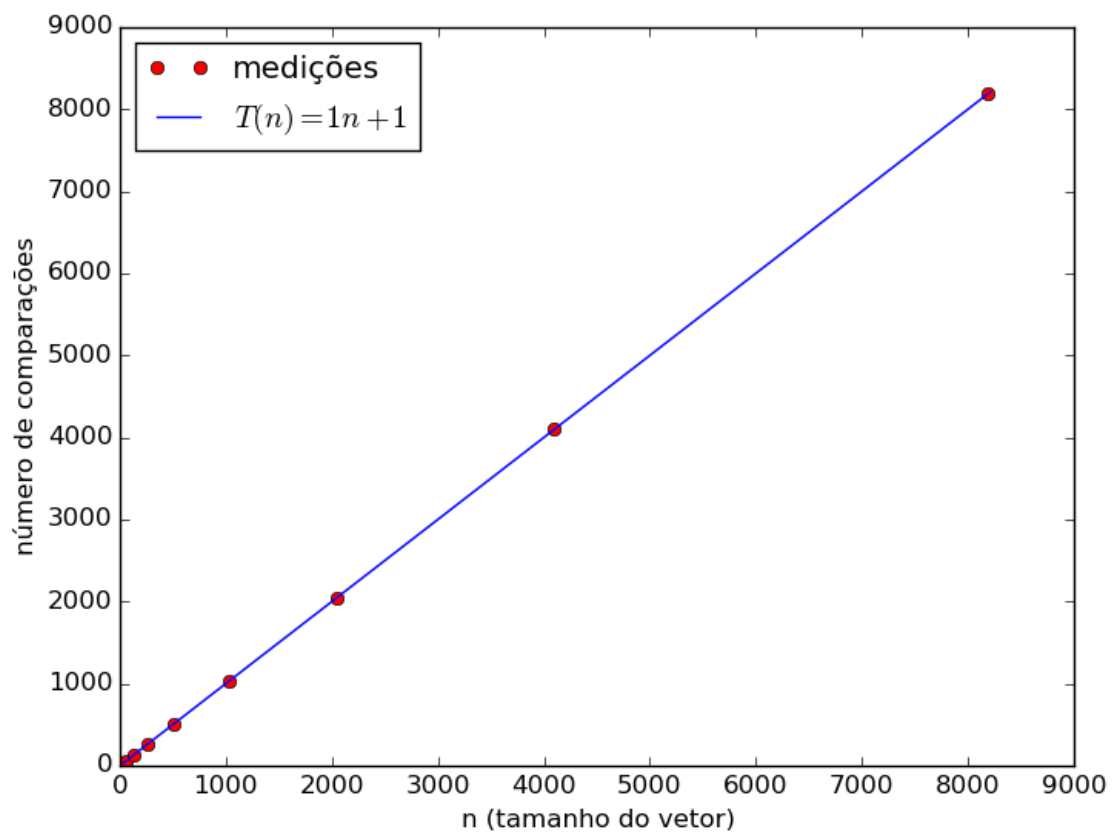


Figura 2.24: A análise do grafico para 2^{32} segue abaixo para countingsort
Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$

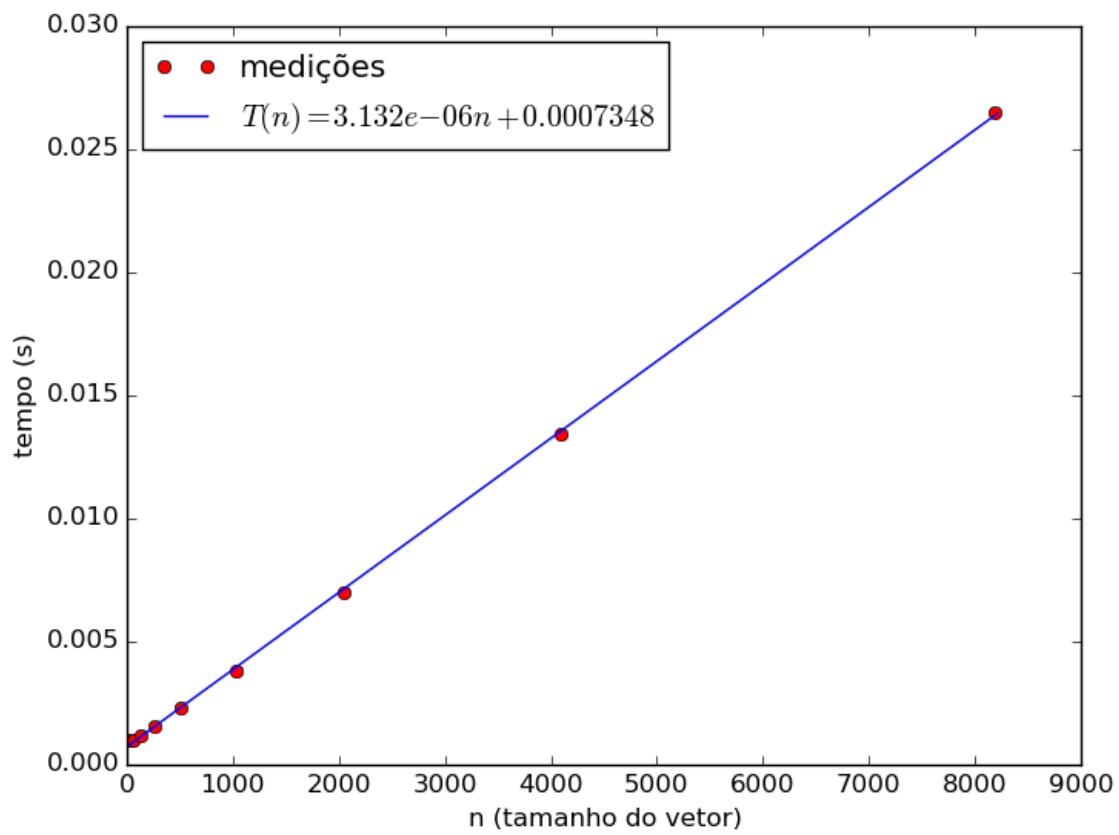


Figura 2.25: A análise do grafico para 2^{32} segue abaixo para counting sort
Tendo a função $T(n) = 3.132e - 6 * n - 0.0007348$ e para o $n = 2^{32}$,
 $T(2^{32}) = 13451.836836272$

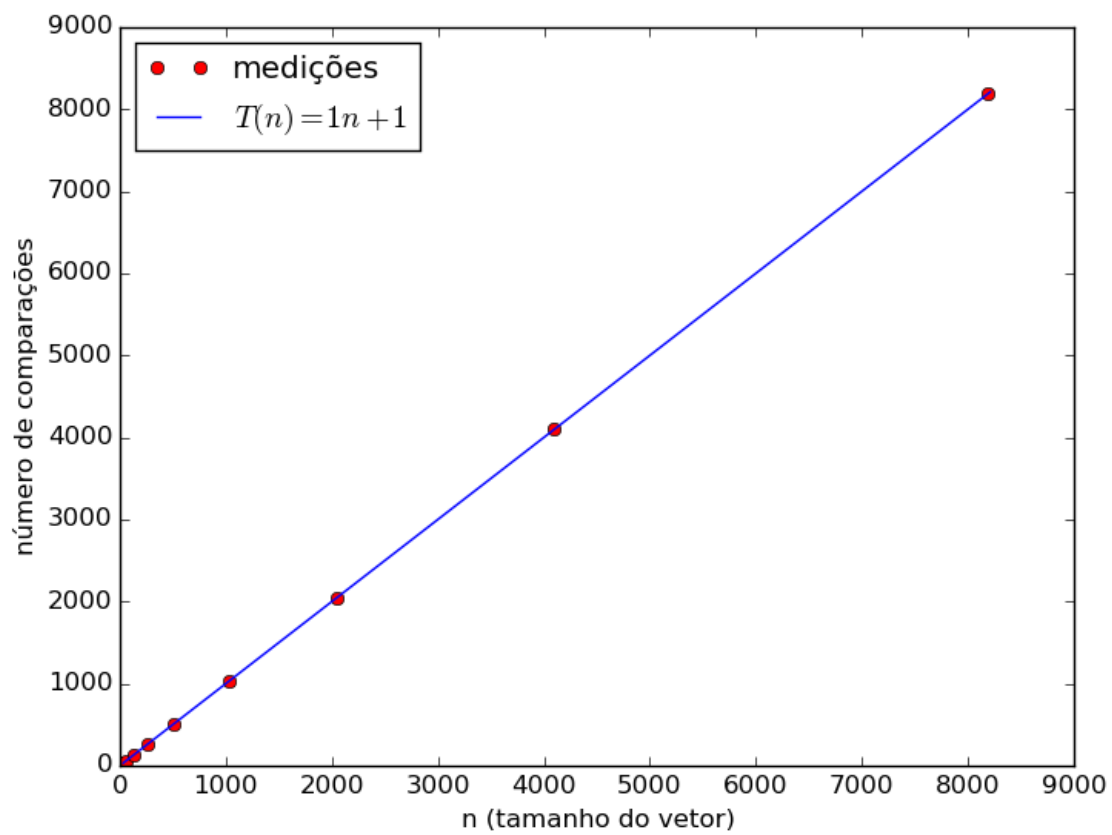


Figura 2.26: A análise do gráfico para 2^{32} segue abaixo para countingsort. Tendo a função $T(n) = n + 1$ e para o $n = 2^{32}$, $T(2^{32}) = 4294967297$.

Apêndice A

Arquivo ../countingsort/countingsort.py

Listagem A.1: ../countingsort/countingsort.py

```
1 @profile
2 def countingsort(aList, k):
3     counter = [0] * ( k + 1 )
4     for i in aList:
5         counter[int(i)] += 1
6     ndx = 0;
7     for i in range( len( counter ) ):
8         while 0 < counter[i]:
9             aList[ndx] = i
10            ndx += 1
11            counter[i] -= 1
```

Apêndice B

Arquivo ../countingsort/ensaio.py

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

```
1 import numpy as np
2 import argparse
3
4 from countingsort 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 countingsort(vet, 1000)
```
