

Emparelhamento estável

Bruno Crotman

25/06/2019

1 Problema do emparelhamento estável

O problema do emparelhamento estável foi estudado em detalhes pela primeira vez em 1962 por Gale e Shapley (Gale and Shapley 1962). O objetivo era gerar um emparelhamento que alocasse alunos a escolas de forma que nenhum aluno a tenha alguma escola e' que ele prefira em comparação à escola e em que ele ficou alocado e ao mesmo tempo essa escola e' prefira a em relação ao aluno a' que ficou alocado originalmente em e' .

Se esta situação ocorrer, o emparelhamento original não é estável, pois basta e' propor a a que a alocação original se desfaz, pois ambos e' e a preferem esta nova alocação.

Este mesmo problema se aplica a inúmeras situações em que há dois grupos de candidatos a uma relação que envolva um membro de cada grupo. Alguns dos exemplos são candidatos a emprego e empresas, pessoas em busca de relacionamento amoroso, calouros de cursos de pós graduação e orientadores, jogadores e clubes, clientes e fornecedores etc. Esta instabilidade na alocação não é desejada pois causa ineficiência: custos de dissolução da relação e custos empregados em treinamento ou qualquer outro tipo de aprofundamento da relação.

Várias aplicações a mercados de combinação deste tipo foram criadas a partir dos algoritmos de Gale e Shapley, como conta Alvin Roth em um de seus livros (Roth 2016). Roth e Shapley dividiram o Nobel de Economia de 2012. Shapley em virtude da descoberta do algoritmo emparelhamento estável, Roth pela aplicação prática de variações deste algoritmo a problemas de emparelhamento diversos como emparelhamento de médicos a hospitais, estudantes com escolas e doadores de órgãos a receptores.

2 Algoritmo Gale-Shapley

O algoritmo implementado foi baseado no livro-texto da matéria (Tardos and Kleinberg 2006). Basicamente o algoritmo segue os seguintes passos, usando um vocabulário de casamentos entre heterossexuais:

- Enquanto há um homem solteiro h
 - h propõe à próxima mulher de sua preferência à qual ainda não propôs, m
 - Se m está solteira
 - * h casa com m
 - Se não
 - * Se ela prefere h ao seu parceiro atual h'
 - m dispensa h' , que fica solteiro, e m casa com h

Para que cada um dos passos seja em tempo constante, é criada uma estrutura de lista para abrigar os homens solteiros e uma matriz para armazenar a posição dos homens no ranking de preferência de cada mulher. A lista possibilita que as operações de inserção e retirada de um solteiro ocorram em $O(1)$. A matriz com o ranking dos homens possibilita que a comparação do parceiro atual da mulher com o homem solteiro candidato ocorra em $O(1)$ também. Com isso, a complexidade do algoritmo é $O(n^2)$.

3 Avaliação da execução do algoritmo

A execução do algoritmo foi avaliada com três tipos de instâncias para 22 diferentes valores de n (número de elementos em cada grupo).

$$n = \{5, 50, 100, 200, \dots, 1000, 1500, \dots, 3000, 4000, \dots, 9000\}$$

Os 3 tipos de instância são os seguintes:

- As que levam ao melhor caso de execução, ou seja, que exigem o menor número possível de passos, para um valor de n ;
- As que levam ao pior caso de execução, ou seja, que exigem o maior número possível de passos, para um valor de n ;
- Instâncias geradas aleatoriamente.

O programa usado para realizar a avaliação empírica do algoritmo foi desenvolvido na linguagem C++, usando apenas as bibliotecas `stdio.h` e `stdlib.h`, para leitura e escrita em arquivos e no console, e `chrono`, para executar a medição dos tempos de execução. Para medição dos tempos de execução foi usado um computador com processador Intel i7-7700 com 3,6 GHz, sistema operacional Windows 10 64 bits. O programa foi compilado no Visual Studio 2017 sem configurações extras de otimização de velocidade.

As instâncias adicionais foram geradas com a linguagem R, que também foi usada para geração deste relatório, juntamente com as bibliotecas `tidyverse` e `RMarkdown`.

3.1 Melhor caso

Na avaliação do melhor caso, foram usadas instâncias específicas, uma para cada valor de n . O algoritmo foi rodado 30 vezes para cada uma das instâncias.

A Figura 1 mostra o número de passos do loop principal do algoritmo (após a preparação das estruturas). É possível ver que o número de passos neste melhor caso é exatamente igual ao número de elementos.

A Figura 2 mostra o tempo em segundos pra cada número de elementos, nas 30 execuções. É possível perceber que o tempo aumenta de forma não linear. A Tabela 1 mostra os coeficientes de uma regressão no formato $tempo = \alpha + n^\beta$, que foi estimada para $n > 1000$. É possível ver que o coeficiente β se aproxima de 2. O tempo da ordem de $O(n^2)$ é justificado porque antes do loop há a inicialização da estrutura do ranking dos homens, que é feito em $O(n^2)$ passos. Como o número de passos mais pesado do loop principal é pequeno, estes passos de inicialização acabam dominando a execução. A Tabela 2 mostra os dados deste melhor caso.

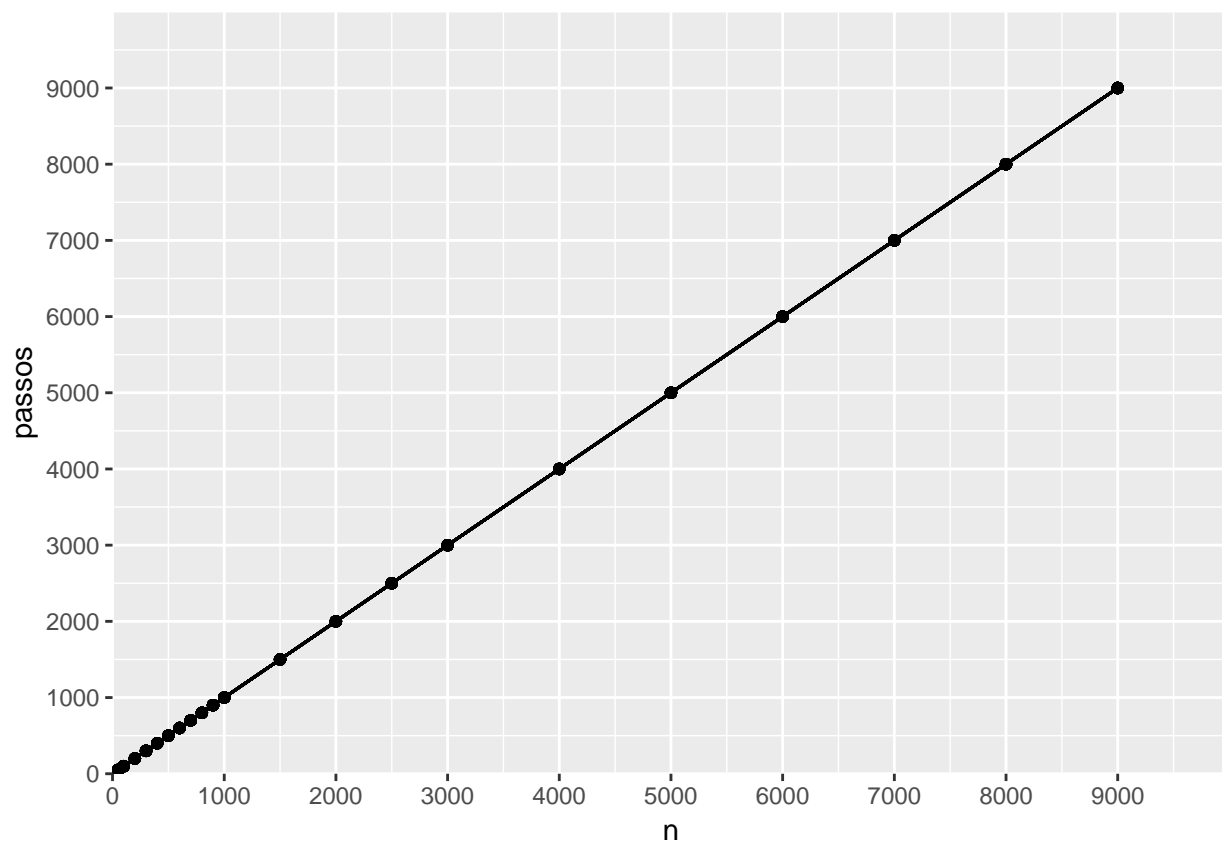


Figura 1: Número de passos x número de elementos em cada grupo

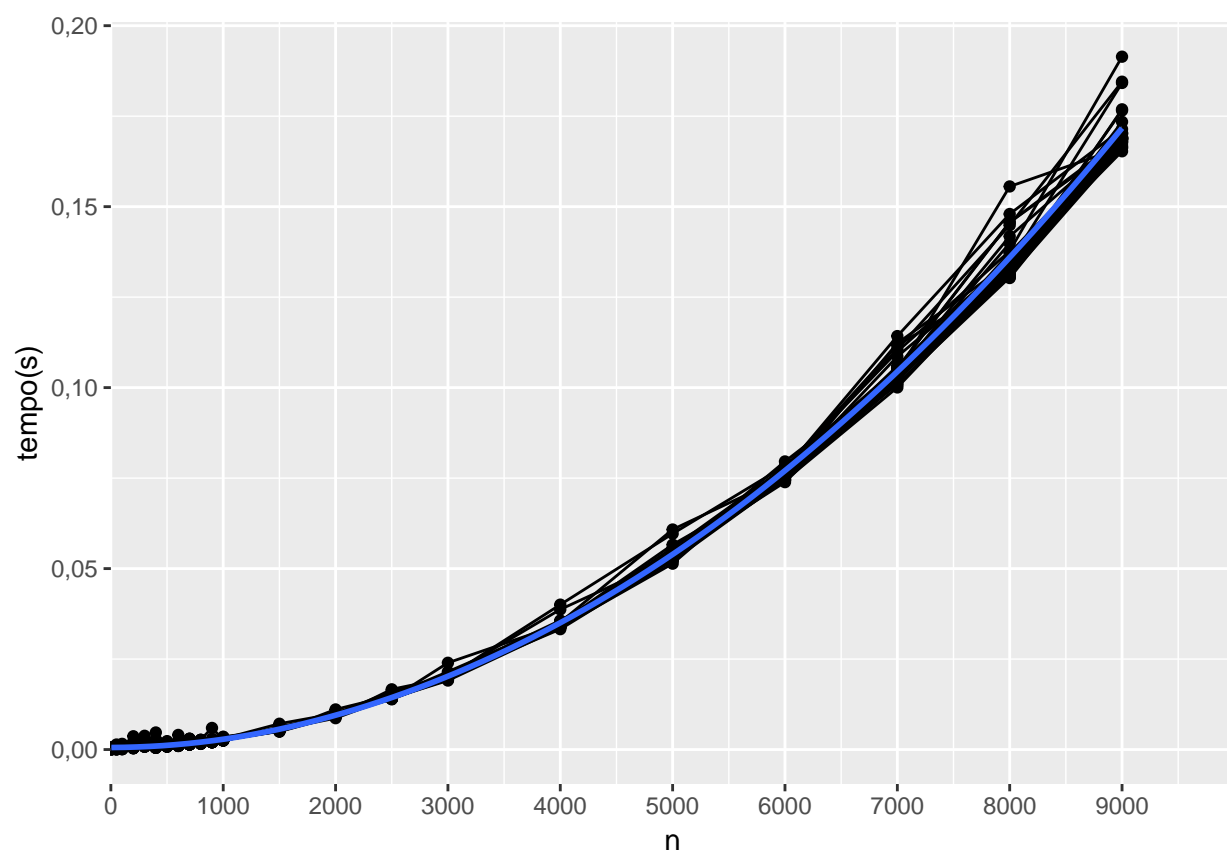


Figura 2: Tempo de execução (segundos) x número de elementos em cada grupo

	Regressão
α	-19,22*** (0,04)
β	1,92*** (0,00)
R^2	1,00
Adj. R^2	1,00
Num. obs.	297
RMSE	0,04

*** $p < 0,001$, ** $p < 0,01$, * $p < 0,05$

Tabela 1: Regressão log-log: $tempo = \alpha + n^\beta$

	n	Tempo médio (ms)	Passos médios	Tempo mediano (ms)	Passos medianos
1	5	0,34	5	0,36	5
2	50	0,41	50	0,34	50
3	100	0,54	100	0,48	100
4	200	0,85	200	0,61	200
5	300	1,29	300	0,99	300
6	400	1,06	400	0,88	400
7	500	1,17	500	1,06	500
8	600	1,31	600	1,14	600
9	700	1,80	700	1,57	700
10	800	1,93	800	1,79	800
11	900	2,32	900	2,14	900
12	1000	2,74	1000	2,65	1000
13	1500	5,52	1500	5,51	1500
14	2000	9,55	2000	9,54	2000
15	2500	14,60	2500	14,32	2500
16	3000	20,21	3000	20,08	3000
17	4000	34,85	4000	34,55	4000
18	5000	54,01	5000	53,56	5000
19	6000	76,46	6000	76,67	6000
20	7000	104,72	7000	103,38	7000
21	8000	136,44	8000	134,57	8000
22	9000	171,34	9000	168,88	9000

Tabela 2: Dados das 30 execuções do melhor caso

3.2 Pior caso

Na avaliação do pior caso, foram usadas instâncias específicas, uma para cada valor de n . O algoritmo foi rodado 30 vezes para cada uma das instâncias.

A Figura 3 mostra o número de passos do loop principal do algoritmo (após a preparação das estruturas). É possível ver que o número de passos neste pior caso evolui de forma não linear. A Tabela 1 mostra os coeficientes de uma regressão no formato $tempo = \alpha + n^\beta$, que foi estimada para $n > 1000$. É possível ver que o coeficiente β é exatamente 2, sugerindo complexidade $O(n^2)$.

A Figura 4 mostra o tempo em segundos pra cada número de elementos, nas 30 execuções. É possível perceber que o tempo aumenta de forma não linear. A Tabela 4 mostra os coeficientes de uma regressão no formato $tempo = \alpha + n^\beta$, que foi estimada para $n > 1000$. É possível ver que o coeficiente β é igual de 2, sugerindo complexidade $O(n^2)$. O coeficiente α , porém, é menos negativo, o que quer dizer que o tempo de execução

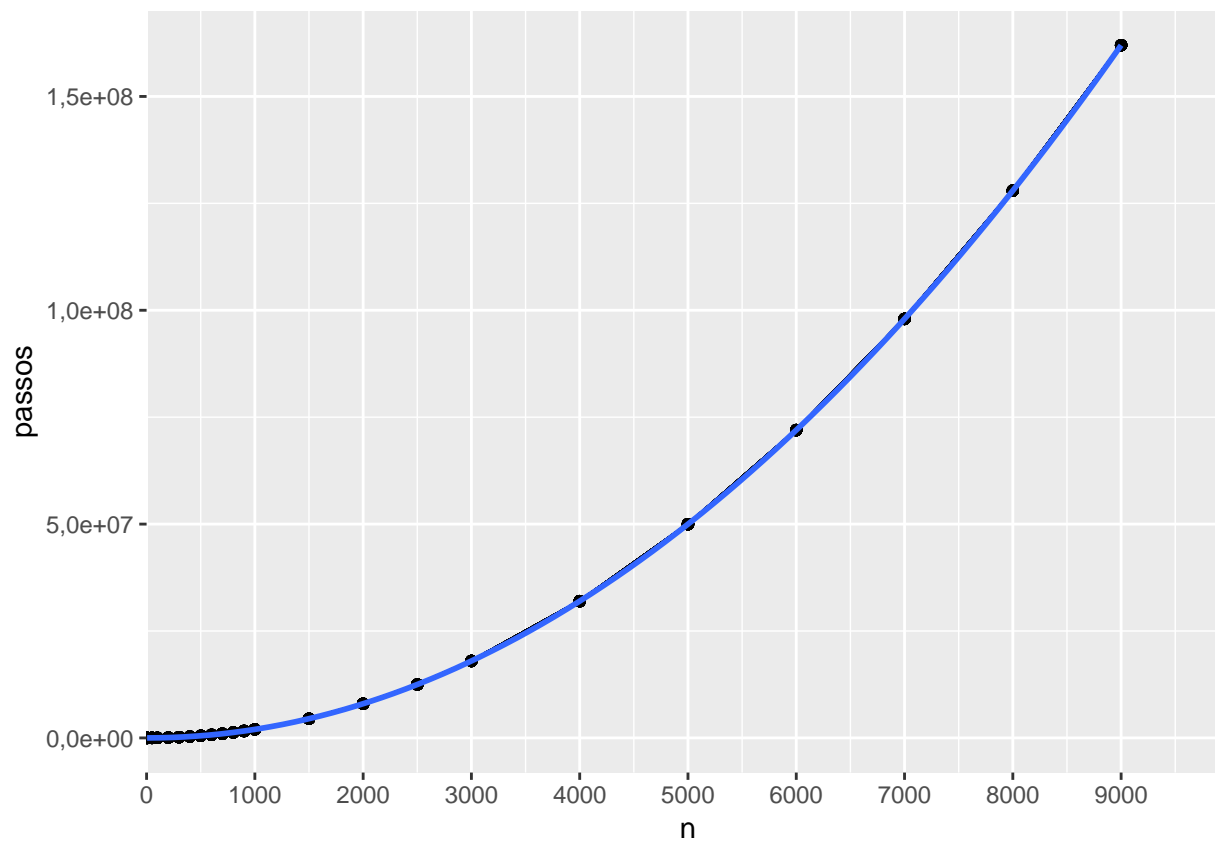


Figura 3: Número de passos x número de elementos em cada grupo

é maior neste caso. Realmente podemos observar que o tempo de execução é bem maior comparando as Tabelas 5 e 2.

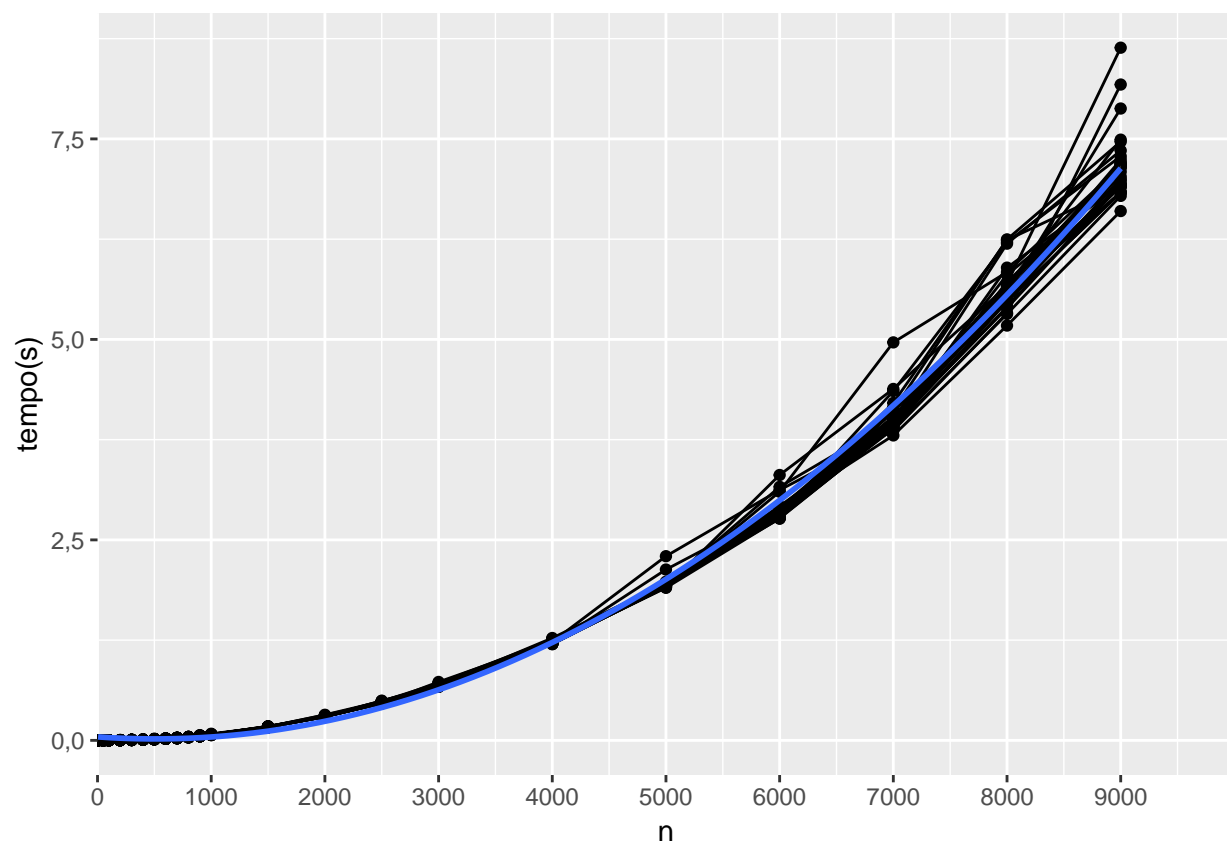


Figura 4: Tempo de execução (segundos) x número de elementos em cada grupo

	Regressão
α	0,69*** (0,00)
β	2,00*** (0,00)
R^2	1,00
Adj. R^2	1,00
Num. obs.	300
RMSE	0,00
*** $p < 0,001$, ** $p < 0,01$, * $p < 0,05$	

Tabela 3: Regressão log-log: $passos = \alpha + n^\beta$

	Regressão
α	-17,13*** (0,04)
β	2,09*** (0,00)
R^2	1,00
Adj. R^2	1,00
Num. obs.	300
RMSE	0,04
*** $p < 0,001$, ** $p < 0,01$, * $p < 0,05$	

Tabela 4: Regressão log-log: $tempo = \alpha + n^\beta$

3.3 Caso médio

Na avaliação do caso médio, foram usadas 30 instâncias geradas aleatoriamente para cada valor de n . O algoritmo foi rodado uma vez para cada uma das instâncias, gerando 30 execuções para cada valor de n . Neste caso, portanto, diferentemente dos anteriores, temos uma variação no número de passos, que passa a não ser mais determinístico.

A Figura 5 mostra o número de passos do loop principal do algoritmo (após a preparação das estruturas). É possível ver que o número de passos neste pior caso evolui de forma quase linear. A Tabela 6 mostra os coeficientes de uma regressão no formato $tempo = \alpha + n^\beta$, que foi estimada para $n > 1000$. É possível ver que o coeficiente β está perto 1,1, sugerindo complexidade do loop principal está perto de $O(n)$ para o caso médio.

A Figura 6 mostra o tempo em segundos pra cada número de elementos, nas 30 execuções. A Tabela 6 relativa à regressão do tempo mostra um coeficiente β perto de 2. É possível perceber que o tempo aumenta de forma não linear. É possível reparar, também, olhando as Tabelas 2, 5 e 8, que o tempo de execução do caso médio está mais perto do tempo do melhor caso do que o do pior caso.

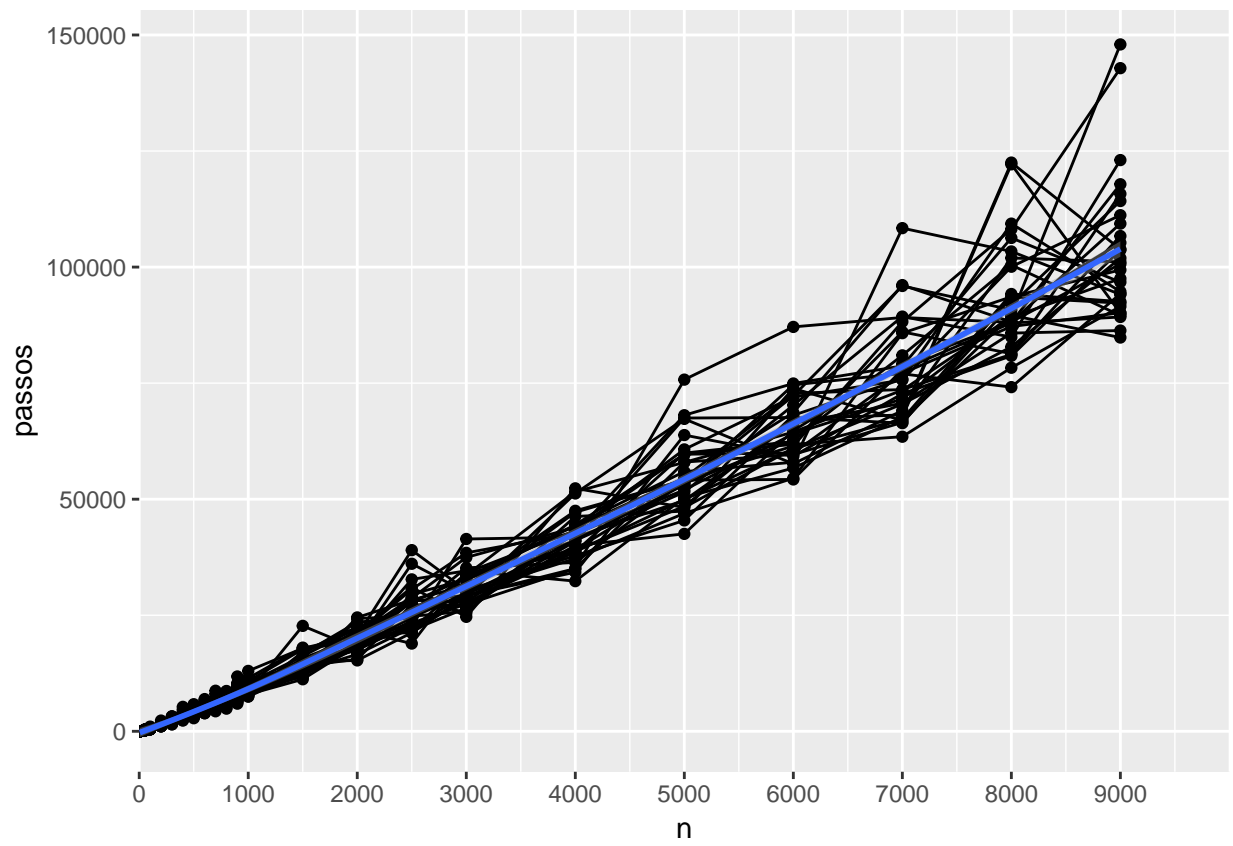


Figura 5: Número de passos x número de elementos em cada grupo

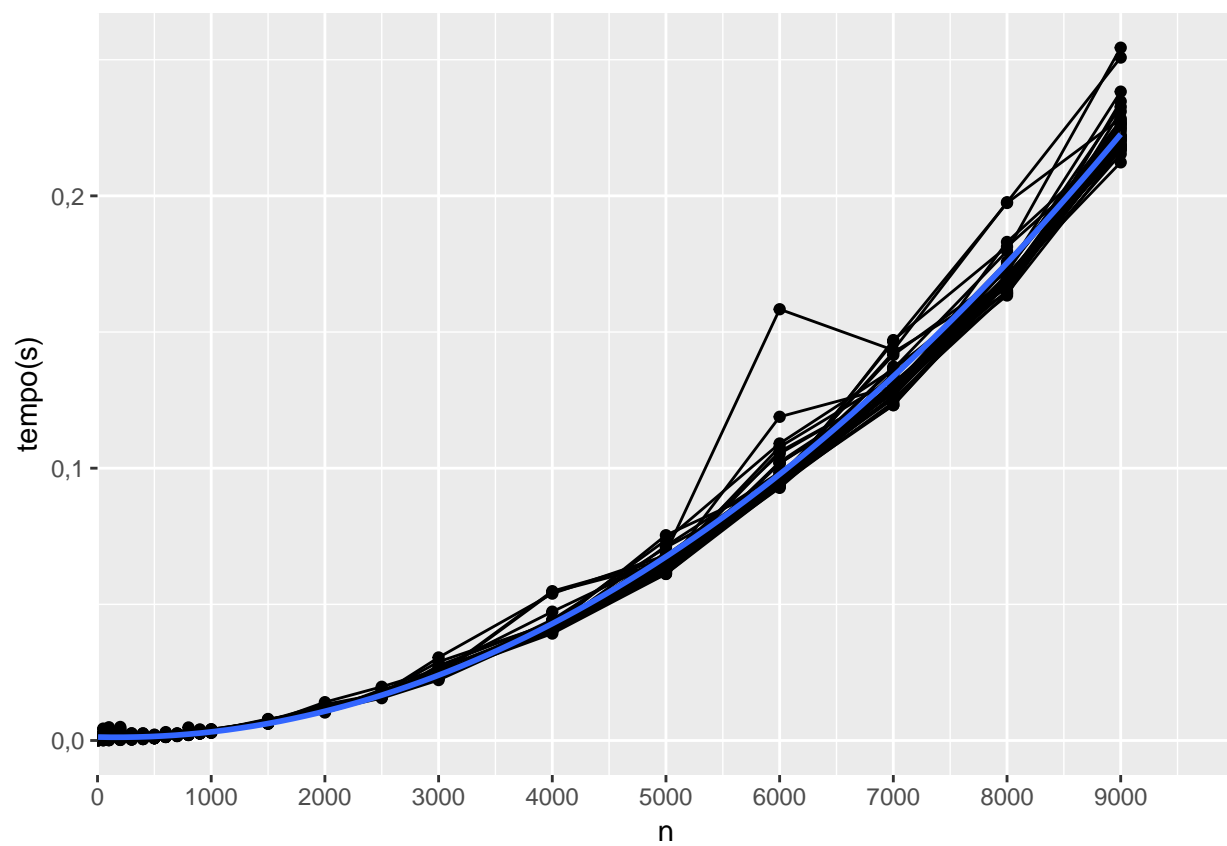


Figura 6: Tempo de execução (segundos) x número de elementos em cada grupo

	n	Tempo médio (ms)	Passos médios	Tempo mediano (ms)	Passos medianos
1	5	0,46	3,70e+01	0,46	3,70e+01
2	50	0,39	4,85e+03	0,32	4,85e+03
3	100	0,71	1,97e+04	0,64	1,97e+04
4	200	2,71	7,94e+04	2,46	7,94e+04
5	300	5,66	1,79e+05	5,52	1,79e+05
6	400	9,34	3,19e+05	9,26	3,19e+05
7	500	14,90	4,99e+05	14,60	4,99e+05
8	600	21,59	7,18e+05	21,47	7,18e+05
9	700	29,41	9,78e+05	29,26	9,78e+05
10	800	40,52	1,28e+06	40,34	1,28e+06
11	900	53,93	1,62e+06	52,83	1,62e+06
12	1000	71,48	2,00e+06	71,35	2,00e+06
13	1500	168,08	4,50e+06	167,34	4,50e+06
14	2000	299,95	7,99e+06	298,92	7,99e+06
15	2500	471,98	1,25e+07	469,82	1,25e+07
16	3000	682,74	1,80e+07	682,12	1,80e+07
17	4000	1227,20	3,20e+07	1223,45	3,20e+07
18	5000	1961,31	5,00e+07	1946,87	5,00e+07
19	6000	2882,05	7,20e+07	2861,26	7,20e+07
20	7000	4049,84	9,80e+07	4005,95	9,80e+07
21	8000	5660,09	1,28e+08	5611,02	1,28e+08
22	9000	7174,14	1,62e+08	7117,24	1,62e+08

Tabela 5: Dados das 30 execuções do pior caso

4 Conclusões

Pudemos atestar que, conforme o esperado, a complexidade do algoritmo se mostra $O(n^2)$ na análise empírica. O tempo de execução segue $O(n^2)$ mesmo no melhor caso e no pior caso, apesar de a execução ser em tempos muito menores do que nas instâncias de pior caso. Isso acontece porque existe uma inicialização antes do loop principal que tem complexidade $O(n^2)$ mesmo no melhor caso e no caso médio. Esta inicialização acaba prevalecendo na análise do tempo assintótica em relação a n .

A descoberta mais interessante foi o fato de que a execução do caso médio está muito mais perto do melhor caso do que do pior caso tanto em número de passos quanto em tempo de execução.

Gale, David, and Lloyd S Shapley. 1962. “College Admissions and the Stability of Marriage.” *The American Mathematical Monthly* 69 (1). Taylor & Francis: 9–15.

Roth, Alvin E. 2016. *Como Funcionam Os Mercados: A Nova Economia Das Combinações E Do Desenho de*

	Regressão
α	1,54*** (0,11)
β	1,10*** (0,01)
R^2	0,96
Adj. R^2	0,96
Num. obs.	300
RMSE	0,13
*** $p < 0,001$, ** $p < 0,01$, * $p < 0,05$	

Tabela 6: Regressão log-log: $passos = \alpha + n^\beta$

	Regressão
α	$-19,36^{***}$ (0,06)
β	$1,96^{***}$ (0,01)
R^2	1,00
Adj. R^2	1,00
Num. obs.	300
RMSE	0,07
*** $p < 0,001$, ** $p < 0,01$, * $p < 0,05$	

Tabela 7: Regressão log-log: $tempo = \alpha + n^\beta$

Mercado. Portfolio-Penguin.

Tardos, Eva, and Jon Kleinberg. 2006. *Algorithm Design*. Reading (MA): Addison-Wesley.

	n	Tempo médio (ms)	Passos médios	Tempo mediano (ms)	Passos medianos
1	5	0,52	1,10e+01	0,49	1,10e+01
2	50	0,91	2,57e+02	0,63	2,45e+02
3	100	0,75	6,05e+02	0,52	5,49e+02
4	200	1,65	1,35e+03	1,13	1,29e+03
5	300	1,12	2,20e+03	1,05	2,08e+03
6	400	1,17	3,20e+03	1,00	3,04e+03
7	500	1,11	4,03e+03	1,00	3,92e+03
8	600	1,71	5,16e+03	1,51	5,05e+03
9	700	2,03	5,94e+03	2,02	5,74e+03
10	800	2,46	6,94e+03	2,31	7,01e+03
11	900	2,93	8,08e+03	2,80	7,96e+03
12	1000	3,41	9,36e+03	3,41	9,33e+03
13	1500	6,80	1,44e+04	6,68	1,40e+04
14	2000	11,59	1,97e+04	11,58	1,92e+04
15	2500	16,96	2,64e+04	16,90	2,57e+04
16	3000	24,47	3,09e+04	23,90	2,99e+04
17	4000	43,31	4,18e+04	42,27	4,15e+04
18	5000	65,08	5,54e+04	63,65	5,40e+04
19	6000	100,47	6,53e+04	96,38	6,43e+04
20	7000	132,21	7,76e+04	129,91	7,58e+04
21	8000	172,48	9,32e+04	169,69	8,96e+04
22	9000	225,10	1,03e+05	222,11	1,00e+05

Tabela 8: Dados das 30 execuções do pior caso