

Análise Experimental de Algoritmos^{1 2}

November 5, 2018

¹David S. Johnson, A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

²Palestra Luciana Buriol, SBPO 2016

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

- Texto metodológico amadurecido ao longo de uma década

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

- Texto metodológico amadurecido ao longo de uma década
- David S. Johnson discute sistematicamente toda a área de Análise Experimental de Algoritmos.

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

- Texto metodológico amadurecido ao longo de uma década
- David S. Johnson discute sistematicamente toda a área de Análise Experimental de Algoritmos.
- Organizado em torno de 10 princípios, são feitas várias sugestões e identificadas as armadilhas (erros mais comuns)

Os 10 princípios Identificados por David S. Johnson

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes
- 6 Garanta a reprodutibilidade

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes
- 6 Garanta a reprodutibilidade
- 7 Garanta a comparabilidade

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes
- 6 Garanta a reprodutibilidade
- 7 Garanta a comparabilidade
- 8 Reporte a história completa

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes
- 6 Garanta a reprodutibilidade
- 7 Garanta a comparabilidade
- 8 Reporte a história completa
- 9 Chegue a conclusões bem-justificadas

Os 10 princípios Identificados por David S. Johnson

- 1 Faça experimentos dignos de nota
- 2 Relacione seu artigo com a literatura
- 3 Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
- 4 Projete os experimentos de forma eficiente e efetiva
- 5 Use implementações suficientemente eficientes
- 6 Garanta a reprodutibilidade
- 7 Garanta a comparabilidade
- 8 Reporte a história completa
- 9 Chegue a conclusões bem-justificadas
- 10 Apresente os dados de modo informativo

- Grupo 1: Princípio 1
 - Grupo 2: Princípios 2, 3, 4, e 5
 - Grupo 3: Princípio 6
 - Grupo 4: Princípio 7, 8 e 9
 - Grupo 5: Princípio 10
-
- Cada grupo deve construir um conjunto de slides com suas conclusões sobre o texto estudado
 - Nos slides devem estar identificados (e comentados) todos os *Pitfalls*, *Pet Peeves* e *Suggestions* citados durante o texto.

Experimentos dignos de nota - Grupo 1: Bruno, Leandro, Robson e José Geraldo

- O artigo do autor Johnson, David S, A Theoretician's Guide to the Experimental Analysis of Algorithms, indica os principais armadilhas e problemas que os pesquisadores não visualizam quando realizam experimentos.
- Simultaneamente apresenta sugestões da melhor maneira de solucionar os mesmos.

Pitfall 1: Lidar com Algoritmos Dominados

- Desperder tempo em algoritmos (ou pesquisas) que já se encontra dominados.

Pet Peeve 1 – Pesquisadores e Revisores que não Fazem o Dever de Casa

- Pesquisadores que citam a pesquisa, mas não lê partes importantes do experimento.

Experimentos dignos de nota (Princípio 1)

- O princípio 1 afirma que os experimentos devem ser dignos de nota.
- Os pesquisadores devem focar em experimentos que tenham uma relevância científica.
- O autor explica que é difícil justificar o uso de experimentos não relevantes, sendo necessário este ser uma alternativa viável.
- Também é de importância científica obter resultados experimentais para modelar fenômenos.

- Os algoritmos estudados podem não ter aplicação prática direta, mas podem ter uma relevância no estudo teórico.

- Os algoritmos é uma alternativa viável para os algoritmos dominados.
- Possíveis Situações que Podem Ocorrer Quando seu Algoritmo é Dominado
 - 1 O algoritmo que domina o seu algoritmo é muito complicado que impede o uso geral.
 - 2 Apesar do algoritmo ser dominado, este possui uma abordagem mais ampla que o algoritmo que o domina.
 - 3 O algoritmo é esperado que tenha um mal desempenho.
 - 4 O algoritmo possui grande interesse individual e sua dominação foi inesperada.

- Deve-se analisar a relevância, credibilidade e generalidade dos resultados da pesquisa.

Pitfall 2 – Dedicar muito tempo computacional para questões erradas

- 1 Estudo excessivo do algoritmo em poucas instâncias.
- 2 Fazer vários experimentos antes de implementar o algoritmo de maneira eficiente.
- 3 Rodar o algoritmo antes de decidir quais dados extrair.

Sugestão 1 – Pense antes de Computar

- Defina os fenômenos que você quer estudar.
- Formule as perguntas que você quer responder.

Sugestão 2 – Use experimentação exploratória para encontrar boas questões

- A experimentação exploratória são testes durante a fase inicial que podem levar a questões relevantes. O autor sugere uma abordagem iterativa:
 - 1 Utilizar metade do tempo de teste procurando anomalias e padrões no seu resultado.
 - 2 Finalizar, então, as implementações do algoritmo, decidir as questões de interesse e realizar experimentos compreensivos para investigá-las.
 - 3 Analisar os resultados. Se as questões não forem respondidas, ou se gerar novas questões, voltar para o passo 2.3. Analisar os resultados. Se as questões não forem respondidas, ou se gerar novas questões, voltar para o passo 2.

Pitfall 3 – Permanecer em um loop infinito de experimentação

- Os experimentos podem ser melhorados, porém é preciso definir um momento para terminar o experimento e deixar possíveis melhorias para as pesquisas futuras. Um artigo digno de notícia que nunca é publicado nunca irá gerar notícias.

Pitfall 3 – Permanecer em um loop infinito de experimentação I

- Questões Relevantes para Responder:
 - 1 Como a escolha dos detalhes de implementação, configuração de parâmetros, heurística e estrutura de dado afetam o desempenho do algoritmo?
 - 2 Como o tempo de execução do algoritmo se relaciona com o tamanho de instância e quão dependente é da estrutura de instância?
 - 3 Quais operações algorítmicas melhor descrevem o tempo de execução?
 - 4 Na prática, quais são os gargalos do algoritmo e como dependem do tamanho e da estrutura da instância? Como isso difere da predição da análise do pior caso?
 - 5 Como o tempo de execução é afetado pela arquitetura da máquina e a utilização de um perfil detalhado ajuda explicar isso?
 - 6 Dado que a pessoa está rodando com instâncias similares e em uma máquina fixa, quão previsível são os tempos de execução?

Pitfall 3 – Permanecer em um loop infinito de experimentação II

- 7 Como o tempo de execução de algoritmo se compara com o dos seus competidores, como essa comparação é afetada pelo tamanho e estrutura da instância e pela arquitetura da máquina e essas diferenças podem ser explicadas em termos de contagem operacional?
- 8 Como são respondidas as perguntas acima quando “tempo de execução” é substituído por “uso de memória” ou o uso de outros recursos computacionais, assumindo que estes usos não são previsíveis com a teoria?
- 9 Quais são as respostas para as perguntas 1,2,6,7 quando a pessoa está utilizando algoritmos de aproximação e “tempo de execução” é substituído por “qualidade da solução”?
- 10 Dado uma classe substancialmente nova de instâncias que você identificou, isso causa uma mudança significativa no comportamento do algoritmo para algoritmos previamente estudados?

Pitfall 4 – Começar usando instâncias geradas aleatoriamente para avaliar o comportamento do algoritmo, porém acaba utilizando o algoritmo para analisar as propriedades das instâncias aleatoriamente geradas.

- Realizando os testes do algoritmo somente em instâncias artificiais pode tornar o trabalho pobre, pois não fica claro como o algoritmo irá funcionar em instâncias do mundo real.
- Realizar experiências a fim de deduzir propriedades de ocorrências aleatórias pode ser justificada por razões matemáticas, e pode interessar mais do que o desempenho de uma heurística.
- No entanto, este é "experimento matemático" e não "análise experimental de algoritmos", e, portanto, foge do sentido que estamos preocupados aqui.