



Estudamos, durante as aulas da disciplina, como a complexidade de tempo (custo de execução) de um algoritmo pode impactar o tempo para a obtenção de uma solução para um problema computacional. O cálculo matemático da complexidade do algoritmo dá uma medida segura para compararmos algoritmos e seus comportamentos.

Neste exercício, procuraremos analisar, na prática, os efeitos de usar estruturas de dados que possuem algoritmos de complexidades diferentes para um mesmo problema. Você realizará a medição de tempo de execução de em cenários com quantidades crescentes de dados e comparará os resultados obtidos com a função de complexidade destes algoritmos. A descrição completa das tarefas segue:

PREPARAÇÃO

- Especificar, no documento a ser entregue, em qual equipamento foram realizados os testes (modelo do processador e velocidade; quantidade de memória principal / RAM; sistema operacional utilizado);
- Observar/entender a classe "Pessoa" disponibilizada no Canvas;
- Observar/entender o programa principal disponibilizada no Canvas. Este programa demonstra o uso de um gerador aleatório para sequências de teste fixas e como medir o tempo de execução de uma tarefa em nanossegundos (10^{-9} segundos);

TAREFAS DO TRABALHO

- Implementar um código que insere um total de P pessoas com id crescente, começando em 1, em uma estrutura *ArrayList* do Java. Após a inserção, este código deve ser capaz de realizar N buscas por pessoas com id's aleatórios na *ArrayList*.
- Implementar um código que insere um total de P pessoas com id crescente, começando em 1, em uma estrutura *HashMap* do Java. Após a inserção, este código deve ser capaz de realizar N buscas por pessoas com id's aleatórios no *HashMap*.
- Para cada implementação acima, executar testes com os valores de P e N abaixo:
 - P = 2.500.000, N = 20.000
 - P = 5.000.000, N = 20.000
 - P = 10.000.000, N = 20.000
- Para cada de (c), marcar o tempo de execução incluindo a criação dos dados e o loop de busca.
- Criar uma tabela comparando os resultados de tempo x tamanho para os resultados obtidos.
- Analisar criticamente esta tabela**, relacionado os resultados obtidos com as ordens de complexidade teóricas conhecidas das estruturas de dados.
- Para as implementações de (a) e (b), executar um teste com P = 2.500.000 e N = 40.000. Marcar o tempo de execução destes testes.
- Analisar criticamente esta último resultado**, debatendo sobre as operações de criação dos dados e busca dos dados nas estruturas utilizadas.

Obs.: pode ser que, em algumas configurações de hardware ou software, você encontre problemas para executar alguns destes cenários de teste. **Não entre em pânico!** Isto é bom sinal; afinal, não estamos estudando a eficiência dos algoritmos e seus consumos de recursos? Se isso acontecer, procure investigar o que está acontecendo para explicar. Se conseguir resolver o problema, explique como fez. Tudo isso faz parte deste tipo de trabalho.

Ressaltando: o documento a ser entregue deve iniciar com as especificações do equipamento e depois, organizadamente e de forma completa deve apresentar os testes. **Mais importante do que os resultados numéricos dos testes** são as explicações relacionando as complexidades e casos encontrados em cada configuração. **Trabalhos que entreguem somente os valores, ou os quais a análise não passa da descrição dos valores ou da reafirmação do que já sabemos (ordem dos algoritmos etc.) ficarão com, no máximo, 30% da nota.**