

TRABALHO PRÁTICO 0
AValiação de Algoritmos**Observações:**

1. Comece a fazer este trabalho imediatamente. Você nunca terá tanto tempo para resolvê-lo quanto agora!
 2. **Data de entrega:** 18 de abril de 2024, até às **23:59 horas**, ou antes.
 3. **Submissão:** Faça a submissão deste trabalho no Moodle, conforme instruções postadas lá.
 4. **Plataforma computacional:** O seu trabalho será executado na plataforma VPL do Moodle.
 5. **Linguagem:** Você deve escrever o seu programa obrigatoriamente na linguagem de programação C++. Não será aceita outra linguagem.
 6. **Documentação:** Veja instruções no Moodle (aguardando a definição do monitor).
 7. **Testes:** O seu programa será avaliado conforme descrito no Moodle da disciplina.
-

Avaliação de Algoritmos

Objetivo do trabalho

Neste trabalho, você vai avaliar algumas soluções de implementação para dois problemas:

1. Ordenação na memória principal
2. Geração da sequência de Fibonacci $(0, 1, 2, 3, 5, 8, 11, \dots)$

Informações importantes. Veja o Moodle da disciplina para informações sobre a submissão.

Definição do problema: Ordenação na memória principal

Descrição. Nesta parte, você deve implementar o Quicksort tanto a versão recursiva quanto a não recursiva. Em ambos os casos, haverá duas possibilidades: a primeira é usar a respectiva versão do Quicksort para fazer a ordenação dos n elementos e a segunda é quando houver partições de no máximo 30 elementos, você deve usar a técnica de ordenação por inserção nessa partição.

O que deve ser feito. Gerar aleatoriamente 50 conjuntos de 10000 números, aplicar as diferentes estratégias de ordenação e calcular os tempos de cada execução. Faça uma análise estatística dos resultados apresentando cada média com o respectivo desvio-padrão. Veja que serão quatro resultados:

- (i) Quicksort recursivo “puro”
- (ii) Quicksort recursivo & Ordenação por inserção
- (iii) Quicksort não recursivo “puro”
- (iv) Quicksort não recursivo & Ordenação por inserção

Definição do problema: Geração da sequência de Fibonacci

Descrição. Nesta parte, você deve gerar os números da sequência de Fibonacci usando tanto a versão recursiva quanto a iterativa.

O que deve ser feito. Tabular os tempos para gerar os números da sequência de Fibonacci em intervalos de 15 segundos até 2 minutos. No livro **Projeto de Algoritmos com Implementações em Pascal e C/em Java e C++** escrito pelo Professor Nivio Ziviani do DCC/UFMG (<https://www2.dcc.ufmg.br/livros/algoritmos/index.php>) há a seguinte tabela que estima o tempo de processamento em uma plataforma real para gerar essa sequência:

n	20	30	50	100
Recursiva	1 s	2 min	21 dias	10^9 anos
Iterativa	1/3 ms	1/2 ms	3/4 ms	1,5 ms

O objetivo é gerar uma tabela informando quantos números da sequência foram gerados usando tanto a versão recursiva quanto a iterativa considerando um tempo de processamento limitado a 15, 30, 45, 60, 75, 90, 105 e 120 segundos.