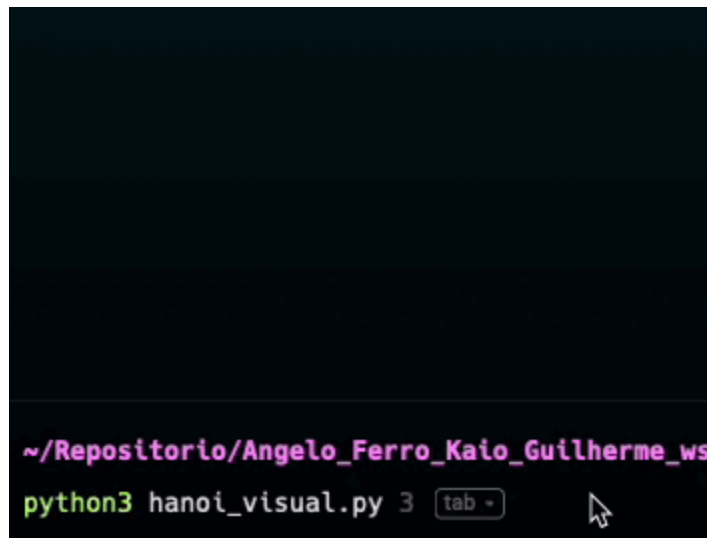


RELATÓRIO: TORRE DE HANOI

Aluno: Angelo Ferro e Kaio Guilherme

1. Introdução

O projeto tem como objetivo analisar o desempenho do algoritmo recursivo da Torre de Hanoi, amplamente utilizado para demonstrar a complexidade de algoritmos exponenciais. A versão implementada foi feita em linguagem C, com medições de tempo e uso de memória via ferramenta Python com psutil.



2. Objetivos

- Avaliar o desempenho da função recursiva da Torre de Hanoi conforme o número de discos aumenta.
- Verificar como a complexidade impacta o tempo de execução e o consumo de recursos.
- Registrar e visualizar o comportamento do sistema em diferentes execuções.

3. Algoritmo Testado

A função hanoi() foi implementada de forma recursiva no arquivo hanoi.c. A função segue o clássico padrão de resolução do problema, movendo n discos da torre de origem para a torre destino, utilizando uma torre auxiliar.

4. Ambiente de Testes

- Linguagem: C (compilado com gcc)
- Sistema Operacional: macOS
- Processador testado: Apple M1 (ARM)
- Ferramentas de medição:
 - Python (benchmark.py com psutil)
- Métricas coletadas:

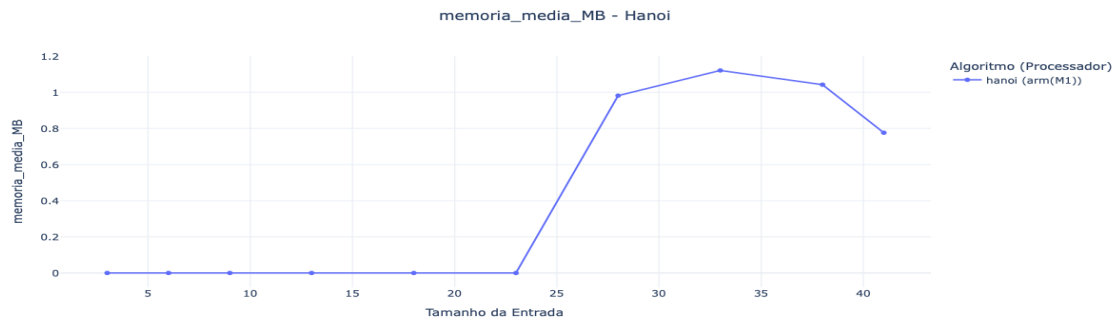
- Tempo médio de execução
- Uso médio de CPU (%)
- Memória média, máxima e mínima (MB)

5. Metodologia

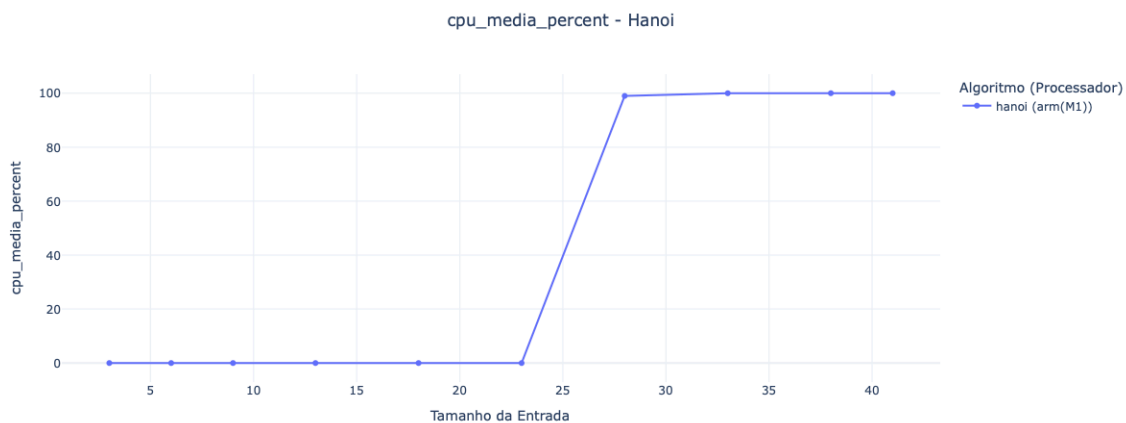
O algoritmo foi executado 13 vezes para cada tamanho de entrada, variando de 10 até 44 discos. Durante cada execução, foram coletadas métricas de uso do sistema. Os dados foram salvos em logs e gráficos foram gerados para visualização de desempenho.

6. Análise dos Resultados

Com o crescimento exponencial do número de chamadas recursivas, o tempo de execução e uso de memória aumentaram drasticamente. A imagem abaixo mostra a média de memória utilizada para 44 discos:

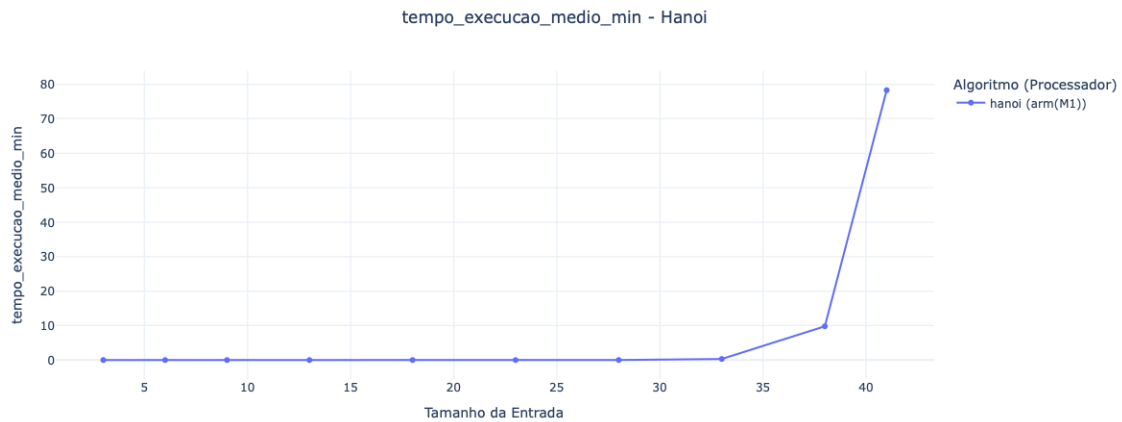


O gráfico abaixo demonstra o uso crescente da CPU ao aumentar o número de discos no algoritmo da Torre de Hanoi. A partir de 28 discos, a CPU atinge 100% de utilização, evidenciando o alto custo computacional da abordagem recursiva.



Este gráfico abaixo apresenta o tempo médio de execução do algoritmo da Torre de Hanoi. Observa-se um crescimento exponencial, com valores que ultrapassam 78

minutos para 40 discos. Isso destaca as limitações práticas do algoritmo recursivo para grandes entradas.



7. Conclusão

- A Torre de Hanoi evidencia bem a complexidade exponencial, com tempo de execução dobrando a cada novo disco.
- O uso de memória aumenta proporcionalmente à profundidade da pilha de chamadas recursivas.

8. Reprodutibilidade

- hanoi.c → algoritmo principal em C.
- benchmark.py → responsável por executar o binário, medir e salvar métricas do sistema.
- Logs e gráficos estão disponíveis nos arquivos de saída do benchmark.