

A estrutura do ep se concentra basicamente na classe principal. Filtro os dados de entrada e os guardo para conseguir processá-los. Uso alguns comandos switch/case para conseguir executar todos os comandos conforme a entrada. As classes de cada estrutura de dado das tabelas de símbolos são separadas em outras classes. Fiz uma limpeza nas palavras tirando pontuações com o uso de regex e deixei tudo em lowercase.

Usei o teste.txt como principal teste de carga, alterando apenas a primeira linha da estrutura de dados. Usei o livro inteiro de iracema, com 24470 palavras. Adiciono tudo de uma vez na tabela de símbolos e depois consulto um select e um rank. Todos retornam o mesmo resultado: alagadas e 2502.

Fiz um arquivo makefile para rodar toda a aplicação. Apenas mude o nome do arquivo de teste dentro do makefile para rodar.

Usei o comando time para medir o tempo de execução. Segue abaixo os tempos de cada execução:

VO: 4.89user 0.18system / 4.09user 0.22system / 4.20user 0.21system / 4.01user 0.16system / 4.51user 0.17system / 4.05user 0.16system / 4.76user 0.21system / 4.21user 0.23system / 4.72user 0.19system / 3.85user 0.18system → **média: 4.33user 0.19system**

ABB: 2.95user 0.17system / 2.95user 0.15system / 3.18user 0.25system / 3.44user 0.19system / 3.69user 0.20system / 3.16user 0.18system / 2.96user 0.19system / 3.00user 0.16system / 3.08user 0.18system / 3.77user 0.21system → **média: 3.22user 0.19system**

TR: 3.11user 0.17system / 3.18user 0.24system / 3.47user 0.15system / 3.38user 0.17system / 3.39user 0.19system / 3.22user 0.14system / 3.06user 0.19system / 3.37user 0.20system / 3.74user 0.17system / 3.53user 0.20system → **média: 3.34user 0.18system**

A23:

ARN: 3.66user 0.22system / 3.06user 0.19system / 3.66user 0.25system / 4.28user 0.22system / 3.63user 0.19system / 3.13user 0.20system / 3.23user 0.15system / 3.29user 0.15system / 3.87user 0.14system / 2.95user 0.18system → **média: 3.48user 0.17system**

Para uma carga pequena de quantidade (teste2.txt) os tempos são muito próximos. O que mostra a diferença apenas para números grandes.

VO: 0.41user 0.06system / 0.43user 0.05system / 0.41user 0.06system / 0.40user 0.07system / 0.42user 0.05system / 0.41user 0.06system / 0.42user 0.05system / 0.42user 0.05system / 0.40user 0.07system / 0.43user 0.05system → **média: 0.41user 0.06system**

ABB: 0.42user 0.06system / 0.44user 0.06system / 0.42user 0.05system / 0.43user 0.04system / 0.41user 0.06system / 0.41user 0.05system / 0.40user 0.07system / 0.42user 0.05system / 0.43user 0.06system / 0.50user 0.07system → **média: 0.43user 0.06system**

TR: 0.45user 0.06system / 0.41user 0.07system / 0.44user 0.05system / 0.42user 0.05system / 0.39user 0.08system / 0.42user 0.05system / 0.42user 0.04system / 0.42user 0.05system / 0.42user 0.06system / 0.42user 0.05system → **média: 0.42user 0.06system**

A23: 0.42user 0.05system / 0.44user 0.07system / 0.45user 0.05system / 0.41user 0.07system / 0.43user 0.05system / 0.55user 0.05system / 0.43user 0.06system / 0.44user 0.05system / 0.41user 0.06system / 0.43user 0.05system → **média: 0.44user 0.06system**

ARN: 0.43user 0.05system / 0.46user 0.03system / 0.55user 0.08system / 0.53user 0.06system / 0.43user 0.05system / 0.40user 0.08system / 0.40user 0.07system / 0.47user 0.07system / 0.40user 0.07system / 0.43user 0.05system → **média: 0.45user 0.06system**

Esses tempos acabam sendo reflexo da análise de tempo de cada estrutura:

| Estrutura de Dados | Busca | Rank | Inserção |
|--------------------|--------------------|--------------------|--------------------|
| VO | $\log n$ | $\log n$ | $\log n + cte$ |
| ABB | $\log n < h < n-1$ | $\log n < h < n-1$ | $\log n < h < n-1$ |
| TR | $\log n$ | $\log n$ | $\log n$ |
| A23 | $\log n$ | h | $\log n$ |
| ARN | $\log n$ | h | $\log n$ |

Como a inserção acaba sendo o maior peso, dado a quantidade de vezes que é executado. É ela quem dita o tempo de execução, basicamente. Podemos ver que o VO por ter uma constante somada, faz um grande impacto. Acaba aumentando 1s do tempo das outras. Todas as outras se mantiveram ali próximo de $\log n$. A ABB poderia chegar até $n-1$, mas nesse caso se comportou em $\log n$. Então a única que demorou um pouco mais foi o VO. Todas as outras estruturas baseadas em árvore foram bem próximas. Então a escolha de cada uma seria baseado em outros fatores, como memória e facilidade de implementação.