

EP2 - Tabela De Símbolos

MAC0323 – Algoritmos e Estruturas de Dados II

Sabrina Araújo da Silva - nºUSP 12566182

Implementação de uma tabela de símbolos usando **vetor ordenado**, **árvore de busca binária**, **treap**, **árvore 2-3** e **árvore rubro-negra**.

Para cada implementação da tabela de símbolos ordenada foram implementadas as seguintes operações:

```
void add (Key key, Item val);
Item value (Key key);
int rank (Key key);
Key select (int k);
```

Para testar o tempo de execução de cada implementação, as quatro operações acima serão testadas com diferentes tipos de texto, sendo eles:

- O livro **The Southern Literary Messenger, Vol. II., No. 3, February, 1836 by Various** disponível no Projeto Gutenberg.
- Texto gerado aleatoriamente com 50 parágrafos, no estilo loren ipsum.

Livro do Projeto Gutenberg

O livro The Southern Literary Messenger (disponível em <https://www.gutenberg.org/ebooks/68141>) escolhido do Projeto Gutenberg possui cerca de 82691 palavras. As seguintes instruções foram feitas para cada uma das 5 implementações:

- 1 82691
- 2 at
- 3 in
- 4 780

*observação 1: o **tempo total** se refere ao tempo total da execução da implementação e não da soma dos tempos das operações. O tempo de **0 segundos** se refere à um tempo muito pequeno em relação aos outros tempos.*

observação 2: para fazer os testes, a saída do programa foi alterada para ter comentários e assim facilitar o entendimento. Exemplo: "A palavra at apareceu no texto 399 vez(es) ate o momento." ao invés de "399". Porém no programa enviado, as saídas estão como no PDF do EP.

Vetor Ordenado

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 82691**

Tempo de execucao para a instrucao 1: 0.947 segundos.

Para a operação `Item value (Key key);` com a instrução **2 at**

A palavra at apareceu no texto 399 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0.001 segundos.

Para a operação `int rank (Key key);` com a instrução **3 in**

8003 palavra(s) sao/eh menor(es) que in.

Tempo de execucao para a instrucao 3: 0 segundos.

Para a operação `Key select (int k);` com a instrução **4 780**

A 780-esima chave da tabela eh Archelais.

Tempo de execucao para a instrucao 4: 0 segundos.

Tempo total

Tempo de execucao para a estrutura VO: 0.962 segundos.

Árvore de Busca Binária

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 82691**

Tempo de execucao para a instrucao 1: 0.157 segundos.

Para a operação `Item value (Key key);` com a instrução **2 at**

A palavra at apareceu no texto 399 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0 segundos.

Para a operação `int rank (Key key);` com a instrução **3 in**

8003 palavra(s) sao/eh menor(es) que in.

Tempo de execucao para a instrucao 3: 0.003 segundos.

Para a operação `Key select (int k);` com a instrução **4 780**

A 780-esima chave da tabela eh Archelais.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura ABB: 0.175 segundos.

Treap

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 82691**

Tempo de execucao para a instrucao 1: 0.185 segundos.

Para a operação `Item value (Key key);` com a instrução **2 at**

A palavra at apareceu no texto 399 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0.001 segundos.

Para a operação `int rank (Key key);` com a instrução **3 in**

8003 palavra(s) sao/eh menor(es) que in.

Tempo de execucao para a instrucao 3: 0.003 segundos.

Para a operação `Key select (int k);` com a instrução **4 780**

A 780-esima chave da tabela eh Archelais.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura TR: 0.201 segundos.

Árvore 2-3

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 82691**

Tempo de execucao para a instrucao 1: 0.06 segundos.

Para a operação `Item value (Key key);` com a instrução **2 at**

A palavra at apareceu no texto 399 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0.001 segundos.

Para a operação `int rank (Key key);` com a instrução **3 in**

8003 palavra(s) sao/eh menor(es) que in.

Tempo de execucao para a instrucao 3: 0.001 segundos.

Para a operação `Key select (int k);` com a instrução **4 780**

A 780-esima chave da tabela eh Archelais.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura ADT: 0.098 segundos.

Árvore rubro-negra

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 82691**

Tempo de execucao para a instrucao 1: 0.056 segundos.

Para a operação `Item value (Key key);` com a instrução **2 at**

A palavra at apareceu no texto 399 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0 segundos.

Para a operação `int rank (Key key);` com a instrução **3 in**

8003 palavra(s) sao/eh menor(es) que in.

Tempo de execucao para a instrucao 3: 0.003 segundos.

Para a operação `Key select (int k);` com a instrução **4 780**

A 780-esima chave da tabela eh Archelais.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura ARN: 0.073 segundos.

Texto no estilo loren ipsum

O texto foi gerado aleatoriamente pelo site <https://mussumipsum.com> e possui 30 parágrafos com cerca de 1100 palavras. As seguintes instruções foram feitas para cada uma das 5 implementações:

- 1 1100
- 2 um
- 3 num
- 4 0

Vetor Ordenado

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 1100**

Tempo de execucao para a instrucao 1: 0.001 segundos.

Para a operação `Item value (Key key);` com a instrução **2 um**

A palavra um apareceu no texto 11 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0 segundos.

Para a operação `int rank (Key key);` com a instrução **3 num**

180 palavra(s) sao/eh menor(es) que num.

Tempo de execucao para a instrucao 3: 0.001 segundos.

Para a operação `Key select (int k);` com a instrução **4 0**

A 0-esima chave da tabela eh A.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura VO: 0.003 segundos.

Árvore de busca binária

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 1100**

Tempo de execucao para a instrucao 1: 0.002 segundos.

Para a operação `Item value (Key key);` com a instrução **2 um**

A palavra um apareceu no texto 11 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0 segundos.

Para a operação `int rank (Key key);` com a instrução **3 num**

180 palavra(s) sao/eh menor(es) que num.

Tempo de execucao para a instrucao 3: 0 segundos.

Para a operação `Key select (int k);` com a instrução **4 0**

A 0-esima chave da tabela eh A.

Tempo de execucao para a instrucao 4: 0 segundos.

Tempo total

Tempo de execucao para a estrutura ABB: 0.006 segundos.

Treap

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 1100**

Tempo de execucao para a instrucao 1: 0.002 segundos.

Para a operação `Item value (Key key);` com a instrução **2 um**

A palavra um apareceu no texto 11 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0.001 segundos.

Para a operação `int rank (Key key);` com a instrução **3 num**

180 palavra(s) sao/eh menor(es) que num.

Tempo de execucao para a instrucao 3: 0 segundos.

Para a operação `Key select (int k);` com a instrução **4 0**

A 0-esima chave da tabela eh A.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura TR: 0.005 segundos

Árvore 2-3

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 1100**

Tempo de execucao para a instrucao 1: 0.001 segundos.

Para a operação `Item value (Key key);` com a instrução **2 um**

A palavra um apareceu no texto 11 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0 segundos.

Para a operação `int rank (Key key);` com a instrução **3 num**

180 palavra(s) sao/eh menor(es) que num.

Tempo de execucao para a instrucao 3: 0 segundos.

Para a operação `Key select (int k);` com a instrução **4 0**

A 0-esima chave da tabela eh A.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura ADT: 0.004 segundos.

Árvore rubro-negra

Saídas:

Para a operação `void add (Key key, Item val);` com a instrução **1 1100**

Tempo de execucao para a instrucao 1: 0.001 segundos.

Para a operação `Item value (Key key);` com a instrução **2 um**

A palavra um apareceu no texto 11 vez(es) ate o momento.

Tempo de execucao para a instrucao 2: 0.001 segundos.

Para a operação `int rank (Key key);` com a instrução **3 num**

180 palavra(s) sao/eh menor(es) que num.

Tempo de execucao para a instrucao 3: 0 segundos.

Para a operação `Key select (int k);` com a instrução **4 0**

A 0-esima chave da tabela eh A.

Tempo de execucao para a instrucao 4: 0.001 segundos.

Tempo total

Tempo de execucao para a estrutura ARN: 0.003 segundos.

Desempenho no total de cada implementação

Texto do Projeto Gutenberg

Operação	VO	ABB	TR	A23	ARN
add	0.947s	0.157s	0.185s	0.06s	0.056s
value	0.001s	0s	0.001s	0.001s	0s
rank	0s	0.003s	0.003s	0.001s	0.003s
select	0s	0.001s	0.001s	0.001s	0.001s
total	0.962s	0.175s	0.201s	0.087s	0.073s

Em ordem de execução mais rápida, temos:

- 1. ARN
- 2. A23
- 3. ABB
- 4. TR
- 5. VO

Texto no estilo loren ipsum

Operação	VO	ABB	TR	A23	ARN
add	0.001	0.002s	0.002s	0.001s	0.001s
value	0s	0s	0.001s	0s	0.001s
rank	0.001s	0s	0s	0s	0s
select	0.001s	0s	0.001s	0.001s	0.001s
total	0.003s	0.006s	0.005s	0.004s	0.003s

Em ordem de execução mais rápida, temos:

- 1. ARN
- 2. VO
- 3. A23
- 4. TR
- 5. ABB