INSTITUTO FEDERAL DO ESPÍRITO SANTO

Rodrigo Couto Rodrigues e João Pedro Garcia Pereira

RELATÓRIO DO TRABALHO DE ÁRVORE BINÁRIA DE BUSCA

Avaliação de Desempenho

Serra 2022

Rodrigo Couto Rodrigues e João Pedro Garcia Pereira

RELATÓRIO DO TRABALHO DE ÁRVORE BINÁRIA

Avaliação de Desempenho

Relatório apresentado no
curso bacharelado
em sistemas de
informação do
instituto federal
do espírito santo

Professor Victorio Albani de
Carvalho

Serra 2022

Sumário

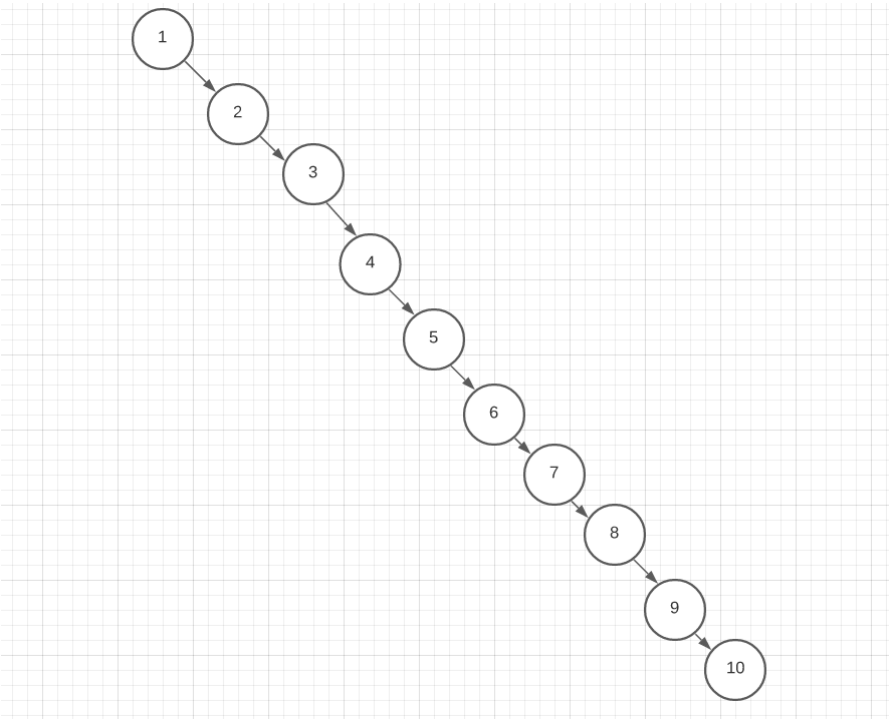
Relato Sobre o Desenvolvimento

A implementação do trabalho foi iniciada com um pouco de facilidade, mas travamos na parte de ser um árvore genérica, não sabíamos prosseguir, tiramos dúvidas na aula, e descobrimos a utilização do Comparable, as lógicas de cada função foram entendidas em sala de aula. Decidimos realizar a implementação do trabalho na linguagem Java, pois a dupla possui mais contato com a linguagem, entre outras derivadas como o dart. Mesmo com a lógica na cabeça, tivemos dificuldades de implementar três métodos, sendo eles, remover um objeto da árvore, obter o pior caso de busca e caminhar em nível. Tivemos dificuldades de remover a raiz da árvore corretamente, e o pior caso de busca não conseguimos implementar de forma correta. O método de caminhar em nível depois de algumas horas desenhando a lógica, conseguimos alcançar uma implementação correta.

Avaliação do Desempenho de Buscas em Árvores Degeneradas

No método ordenado notamos que a topologia da árvore se torna uma lista encadeada ou a um vetor, o pior caso de busca será N.

Um exemplo feito no Lucidchart para demonstrar a topologia da árvore ordenada:



Sendo assim, a complexidade do pior caso é igual à altura da árvore, todos os elementos são percorridos no pior caso. O pior caso pode ser O(n), onde n é o número de nós da árvore.

Quantidade de elementos: 10

```
##--------------------Menu-------------------##

|-------------------------------------------|
| Opção 1 - Exibir estatísticas             |
| Opção 2 - Efetuar busca por matrícula     |
| Opção 3 - Excluir por matrícula           |
| Opção 4 - Incluir aluno                   |
| Opção 5 - Sair                            |
|-------------------------------------------|
Digite uma opção: 2

Informe a matrícula do aluno procurado:
2000000010
quantidade de elementos percorridos: 9 - 2000000010;Ipim Acujabe;96.0

Tempo Total de busca do aluno em ms: 1

Pressione Enter para continuar...
```

Quantidade de elementos: 200000

```
##--------------------Menu-------------------##

|-------------------------------------------|
| Opção 1 - Exibir estatísticas             |
| Opção 2 - Efetuar busca por matrícula     |
| Opção 3 - Excluir por matrícula           |
| Opção 4 - Incluir aluno                   |
| Opção 5 - Sair                            |
|-------------------------------------------|
Digite uma opção: 2

Informe a matrícula do aluno procurado:
2000200000
quantidade de elementos percorridos: 199999 - 2000200000;Uujaib Lesae;86.0

Tempo Total de busca do aluno em ms: 16

Pressione Enter para continuar...
```

Quantidade de elementos: 400000

```
##--------------------Menu------------------##

|-------------------------------------------|
| Opção 1 - Exibir estatísticas             |
| Opção 2 - Efetuar busca por matrícula     |
| Opção 3 - Excluir por matrícula           |
| Opção 4 - Incluir aluno                   |
| Opção 5 - Sair                            |
|-------------------------------------------|
Digite uma opção: 2

Informe a matrícula do aluno procurado:
2000400000
quantidade de elementos percorridos: 399999 - 2000400000;Exe Xinodeco;88.0

Tempo Total de busca do aluno em ms: 16

Pressione Enter para continuar...
```

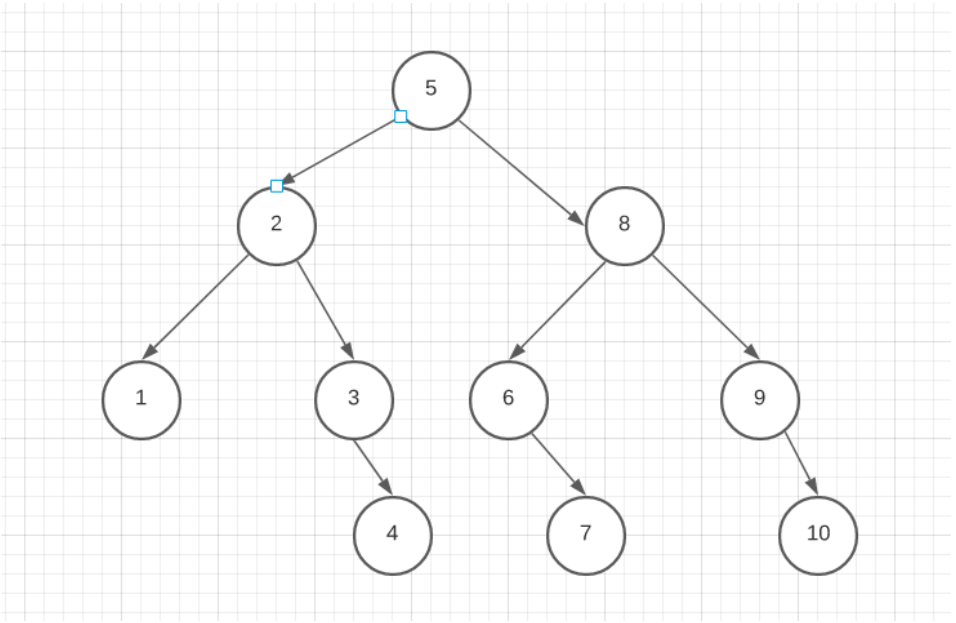
Quantidade de elementos: 600000

```
##--------------------Menu------------------##

|-------------------------------------------|
| Opção 1 - Exibir estatísticas             |
| Opção 2 - Efetuar busca por matrícula     |
| Opção 3 - Excluir por matrícula           |
| Opção 4 - Incluir aluno                   |
| Opção 5 - Sair                            |
|-------------------------------------------|
Digite uma opção: 2

Informe a matrícula do aluno procurado:
2000600000
quantidade de elementos percorridos: 599999 - 2000600000;Koo Rejuc;53.0

Tempo Total de busca do aluno em ms: 16

Pressione Enter para continuar...
```

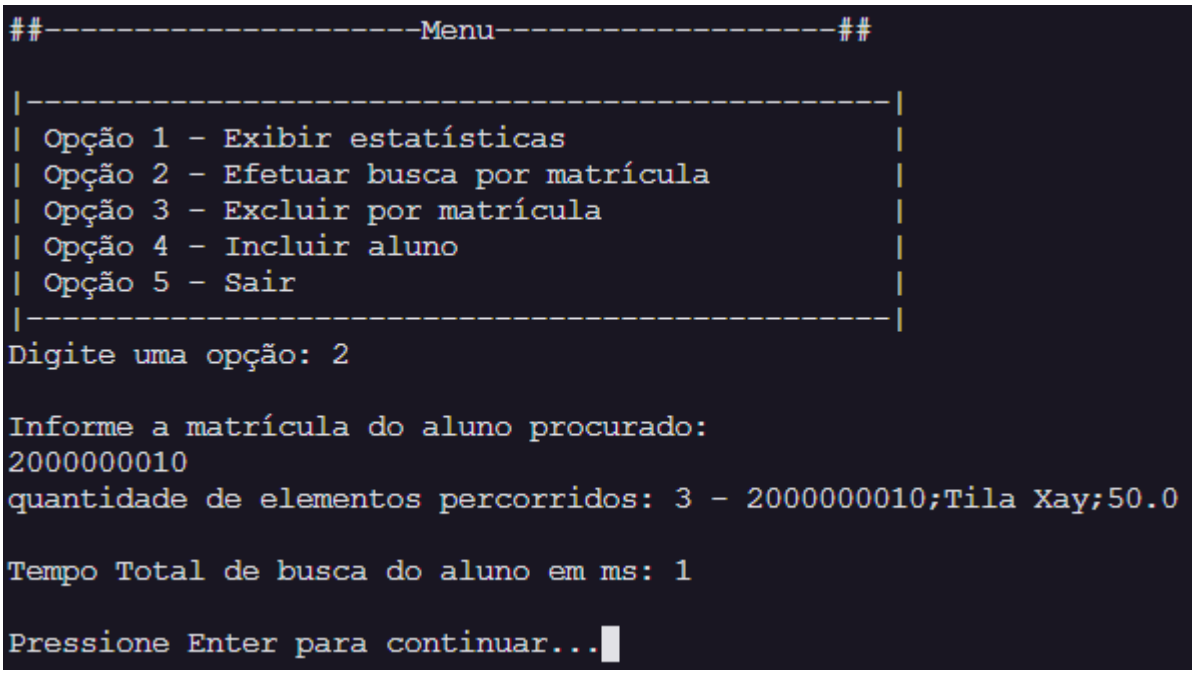
Avaliação do Desempenho de Buscas em Árvores Balanceadas

No método balanceado notamos que a topologia da árvore se torna bem mais eficiente e o pior caso é O(logn).

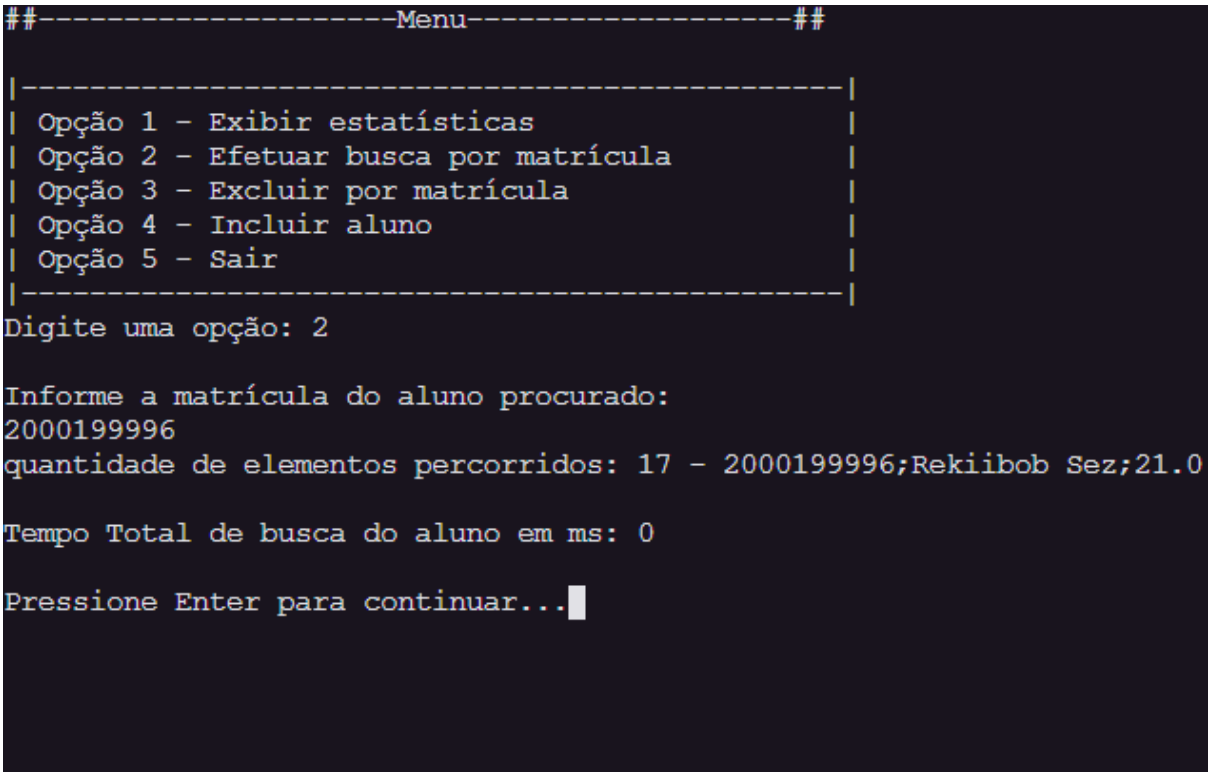
Um exemplo feito no Lucidchart para demonstrar a topologia da árvore balanceada:



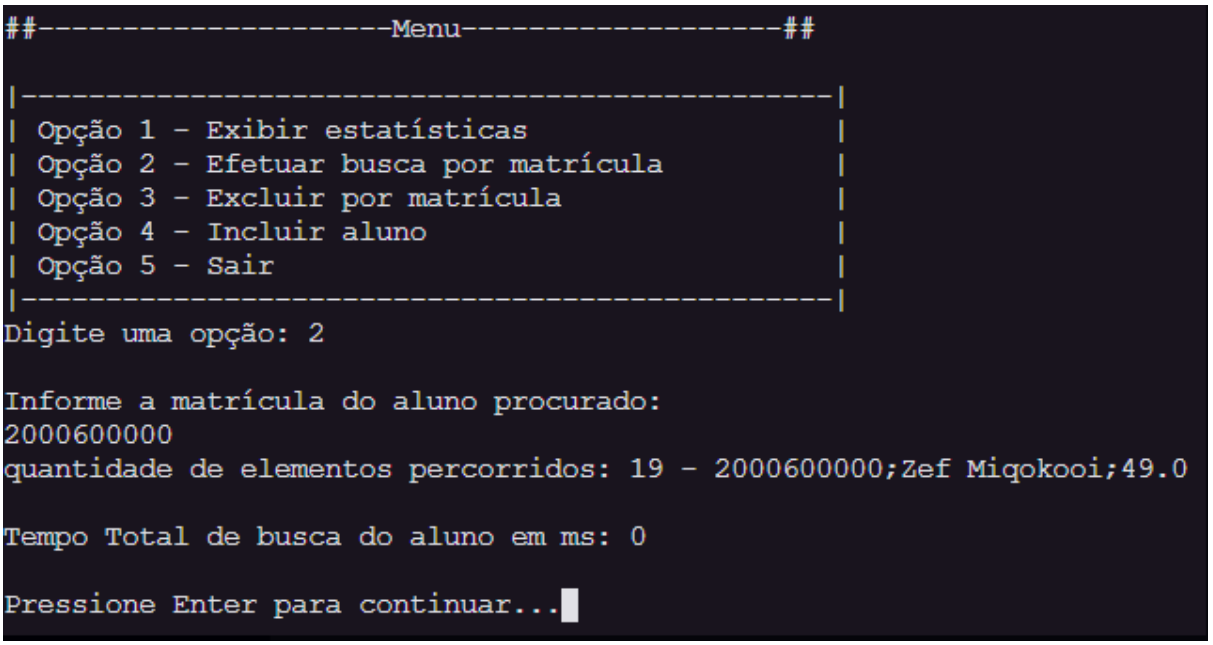
Quantidade de elementos: 10



Quantidade de elementos: 200000

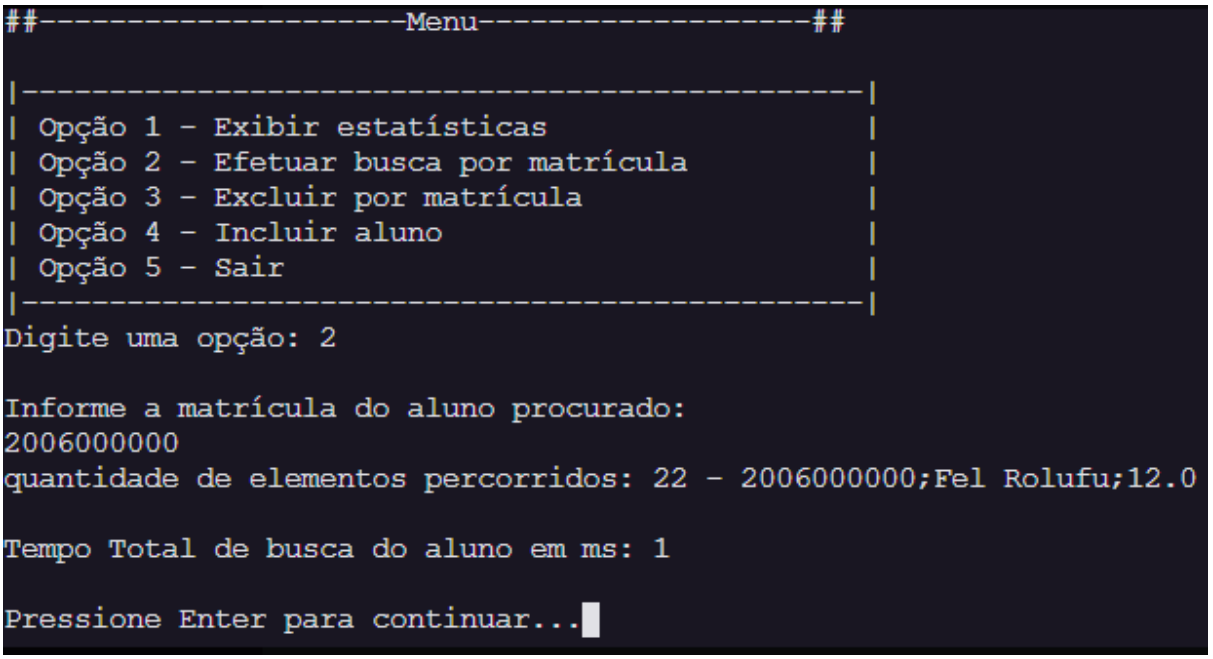


Quantidade de elementos: 600000



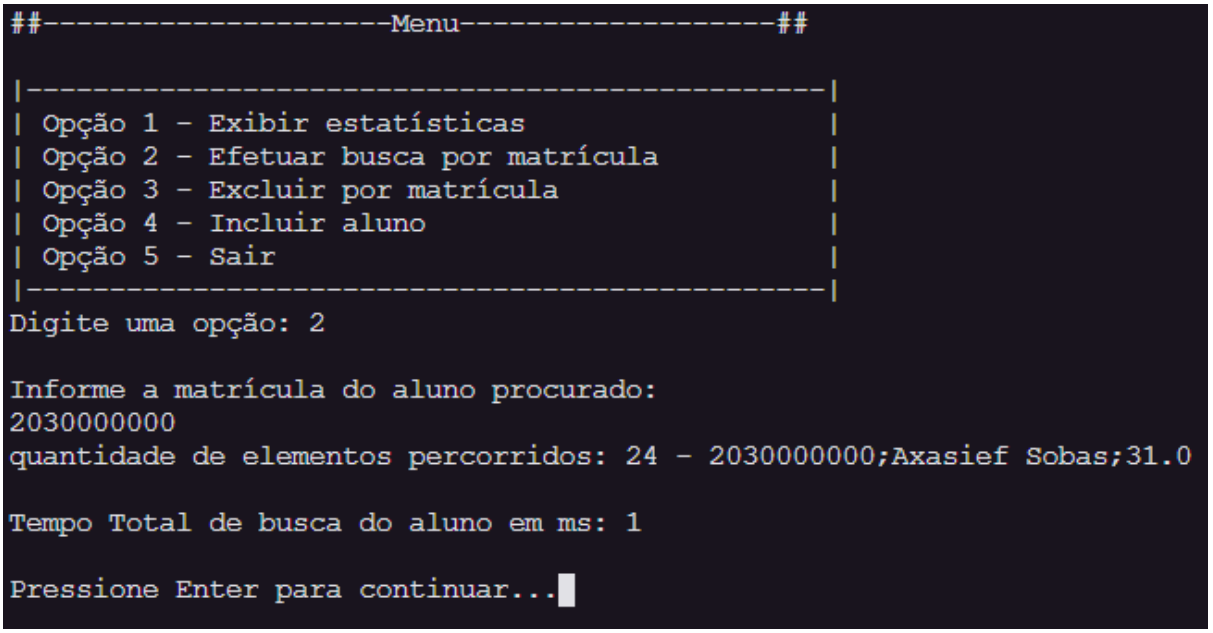
Notamos que mesmo aumentando a quantidade de elementos, não muda o tempo de busca, provando que a árvore balanceada é muito mais eficiente. Vamos tentar aumentar consideravelmente agora…

Quantidade de elementos: 6000000



Aumentamos para 6 milhões de elementos e o tempo total de busca foi para 1ms. Sendo percorrido apenas 22 elementos, na topologia de árvore ordenada teria sido percorrido 5,99 milhões de elementos, e um tempo de busca elevadíssimo.

Quantidade de elementos: 30000000



Mesmo aumentando consideravelmente o tempo de busca ficou com 1ms, com 30 milhões de elementos na árvore.

Avaliação do Tempo de Carga dos Arquivos(Geração das Árvores)

O ordenado tem um tempo de carregamento muito superior em relação ao balanceado, em casos de arquivos pequenos, não é possível notar a diferença, mas no caso do arquivo com 200 mil elementos, o tempo de carregamento do ordenado traz a ideia que o programa travou. Como no caso do ordenado ele precisa percorrer todos os elementos para inserir um novo, isso torna o processo bem demorado. Já no método balanceado, o número de elementos percorridos é bem menor, otimizando o carregamento. No último teste do arquivo ordenado de 600 mil elementos o tempo gasto foi muito alto (+10 minutos), colocamos um de 30 milhões no método balanceado e foi quase que instantâneo.