

Aplicação de uma árvore AVL para gerar um banco de dados

Átila Alves Rodrigues¹ and Samuel Linhares Nogueira¹

¹Centro Universitário do Instituto de Educação Superior de Brasília (IESB)Brasília -
DF - Brazil

Resumo

O tema do projeto integrador foi pesquisar os requisitos para se criar uma carteira de identidade e assim simular um banco de dados, que tenha no mínimo o número populacional de 20 países, é no final fazer uma busca dentro do sistema. O sistema implementado será árvore AVL, será feito a operação de adicionar, procurar e verificar o número de elementos.

abstract

The theme of the integrative project was to meet the requirements to create an identity card and also to simulate a database, which must have a minimum populational number of 20 countries, and the database be able to search within the system. The implemented system will be the AVL Tree, it will have operation of adding, Search and checking the number of elements.

1 Introdução

A fim de simular um ambiente com um grande volume de informações, o projeto em questão criará uma estrutura de árvore balanceada (árvore AVL), assim visando o desempenho nas operações de inserção e busca.

2 Objetivos

2.1 Objetivos gerais

Ao aplicar as operações de uma árvore AVL em um ambiente com 7 bilhões de volume, o principal problema gerado seria o tempo gasto para cada operação, sendo assim o melhor método para solução é a separação desses 7 bilhões de informações. Outro método que ajudaria seria a utilização de uma linguagem em baixo nível.

2.2 Objetivos específicos

- Fazer a implementação da árvore AVL
- Criar uma árvore para cada país
- Gerar um banco de dados a partir das árvores AVL
- Fazer a operação de busca dentro do banco de dados

3 Referencial Teórico

3.1 Árvores Binárias

[Wik]As árvores binárias de pesquisa são projetadas para um acesso rápido à informação. Idealmente a árvore deve ser razoavelmente equilibrada e a sua altura será dada (no caso de estar completa) por $h = \log_2(n+1)$

As árvores completas minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas. Contudo, para garantir essa propriedade em aplicações dinâmicas, é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (inclusão ou exclusão).

3.2 árvore AVL

[Wik]A AVL (Adelson-Velskii e Landis – 1962) é uma árvore altamente balanceada, isto é, nas inserções e exclusões, procura-se executar uma rotina de balanceamento tal que as alturas das sub-árvores esquerda e sub-árvores direita tenham alturas bem próximas (onde cada nó difere no máximo por uma unidade).

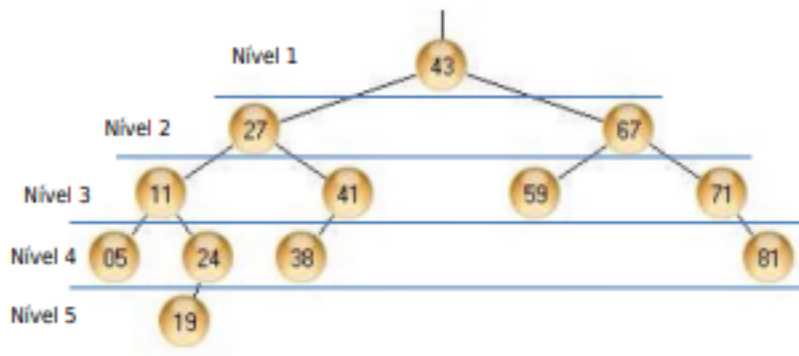


Figura 1: árvore AVL

3.2.1 Operação de Busca

[Wik]A busca em AVL é a mesma utilizada em árvore binária de busca, se procura pela chave de valor K inicia sempre pelo nó raiz da árvore. Seja pt_u um ponteiro para o nó u sendo verificado. Caso o pt_u seja nulo então a busca não foi bem sucedida (K não está na árvore ou árvore vazia). Verificar se a chave K igual $pt_u \rightarrow chave$ (valor chave armazenado no nó u), então a busca foi bem sucedida. Caso contrário, se $K < pt_u \rightarrow chave$ então a busca segue pela subárvore esquerda; caso contrário, a busca segue pela subárvore direita.

3.2.2 Operação de Inserção

[Wik]Para inserir um novo nó de valor K em uma árvore AVL é necessária uma busca por K nesta mesma árvore. Após a busca o local correto para a inserção do nó K será em uma subárvore vazia de uma folha da árvore. Depois de inserido o nó, a altura do nó pai e de todos os nós acima deve ser

atualizada. Em seguida o algoritmo de rotação simples ou dupla deve ser acionado para o primeiro nó pai desregulado.

3.2.3 Operação de Remoção

[Wik]O primeiro passo para remover uma chave K consiste em realizar uma busca binária a partir do nó raiz. Caso a busca encerre em uma subárvore vazia, então a chave não está na árvore e a remoção não pode ser realizada. Caso a busca encerre em um nó u o nó que contenha a chave então a remoção poderá ser realizada da seguinte forma:

Caso 1: O nó u é uma folha da árvore, apenas exclui-lo.

Caso 2: O nó u tem apenas uma subárvore, necessariamente composta de um nó folha, basta apontar o nó pai de u para a única subárvore e excluir o nó u.

Caso 3: O nó u tem duas subárvores: localizar o nó v predecessor ou sucessor de K, que sempre será um nó folha ou possuirá apenas uma subárvore; copiar a chave de v para o nó u; excluir o nó v a partir da respectiva subárvore de u.

O último passo consiste em verificar a desregulagem de todos nós a partir do pai do nó excluído até o nó raiz da árvore. Aplicar rotação simples ou dupla em cada nó desregulado.

3.2.4 Operação de Rotação

[Wik]A operação básica em uma árvore AVL geralmente envolve os mesmos algoritmos de uma árvore de busca binária desbalanceada. A rotação na árvore AVL ocorre devido ao seu desbalanceamento, uma rotação simples ocorre quando um nó está desbalanceado e seu filho estiver no mesmo sentido da inclinação, formando uma linha reta. Uma rotação-dupla ocorre quando um nó estiver desbalanceado e seu filho estiver inclinado no sentido inverso ao pai, formando um "joelho".

Para garantirmos as propriedades da árvore AVL rotações devem ser feitas conforme necessário após operações de remoção ou inserção. Seja P o nó pai, FE o filho da esquerda de P e FD o filho da direita de P podemos definir 4 tipos diferentes de rotação.

3.2.5 Rotação simples à direita

[Wik]Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a 2 e a diferença das alturas h dos filhos de FE é igual a 1. O nó FE deve tornar o novo pai e o nó P deve se tornar o filho da direita de FE. Segue pseudocódigo:



Figura 2: Rotação simples à direita

3.2.6 Rotação simples à esquerda

[Wik] Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a -2 e a diferença das alturas h dos filhos de FD é igual a -1 . O nó FD deve tornar o novo pai e o nó P deve se tornar o filho da esquerda de FD . Segue pseudocódigo:



Figura 3: Rotação simples a esquerda

3.2.7 Rotação dupla à direita

[Wik] Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a 2 e a diferença das alturas h dos filhos de FE é igual a -1 . Nesse caso devemos aplicar uma rotação à esquerda no nó FE e, em seguida, uma rotação à direita no nó P .

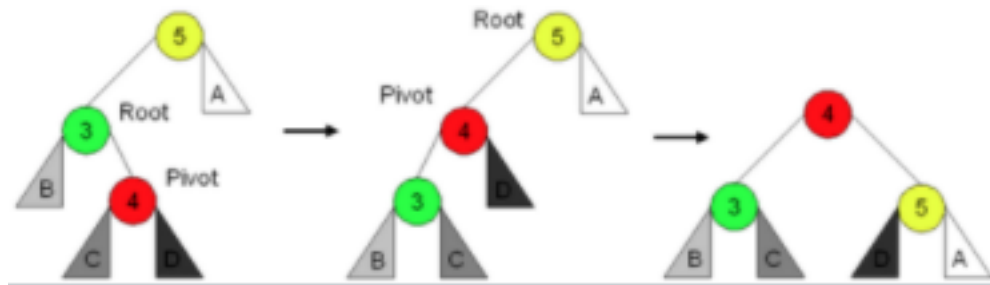


Figura 4: Rotação dupla à direita

3.2.8 Rotação dupla à esquerda

[Wik] Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a -2 e a diferença das alturas h dos filhos de FD é igual a 1 . Nesse caso devemos aplicar uma rotação à direita no nó FD e, em seguida, uma rotação à esquerda no nó P .

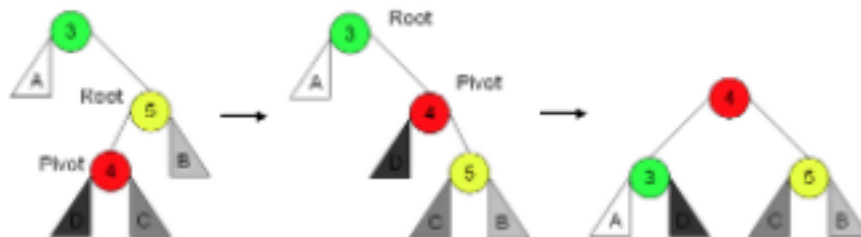


Figura 5: Rotação dupla à esquerda

3.3 Linguagem C

[Nol]Criada pelo cientista da computação Dennis Ritchie, em 1972, a linguagem C foi derivada de outras duas: a BCPL e a Algol 68. Embora tenha sido pensada com o propósito exclusivo de ser usada no desenvolvimento de uma nova versão do sistema operacional Unix, hoje é aplicada nos mais variados tipos de projeto.

Considerada uma linguagem de alto nível genérica, a C pode ser usada em diversos tipos de projeto, como a criação de aplicativos, sistemas operacionais, drivers, entre outros. Trata-se de uma linguagem estruturada que se tornou muito popular nos anos 80 — tanto que é difícil encontrar arquiteturas para as quais não existam compiladores para a C, o que garante o seu elevado nível de portabilidade.

Uma das grandes vantagens dessa linguagem é a capacidade de gerar códigos rápidos, ou seja, com um tempo de execução baixo. Além disso, a programação em C é bastante simplificada, pois sua estrutura é simples e flexível. Tendo isso em mente, podemos dizer que as principais características da linguagem C são:

- Portabilidade;
- Geração de código eficiente;
- Simplicidade;
- Confiabilidade;
- Facilidade de uso;
- Regularidade.

3.3.1 Sintaxe da Linguagem C

[Nol]A sintaxe define quais são as regras para a construção de um programa em uma determinada linguagem. Na C, assim como nas demais linguagens, tais regras estão relacionadas aos tipos, funções e declarações:

- Tipos: definem as propriedades dos dados que serão manipulados em um código;
- Funções: indicam as ações que serão executadas pelo programa quando o código estiver rodando;
- Declarações: são trechos do programa que servem para alocar memória, definir funções, declarar variáveis, etc.

3.3.2 Identificadores

[Nol]Identificador é nome dado aos termos que são usados para nomear uma variável, constante, função, etc. Para definir um identificador, é preciso seguir algumas regras. Veja:

- Deve ser iniciado por uma letra ou pelo símbolo “_”; não se deve usar espaço entre os termos;
- Como C é uma linguagem case-sensitive, ela diferencia letras maiúsculas das minúsculas. Isso significa que um identificador definido como “numero” é diferente de um definido como “Numero”;
- Não se pode usar caracteres especiais ou acentuação;
- É uma boa prática usar nomes que estejam relacionados ao contexto do programa.

Exemplos de identificadores:

- Nome;
- NomeDoAluno;
- PesoDoProduto.

3.3.3 Variáveis

[Nol] Variáveis são, basicamente, uma posição na memória da máquina que identificamos por um nome (identificador). Na linguagem C, todas as variáveis precisam ser declaradas no início do programa, antes de serem usadas.

Elas também devem receber um tipo, que vai determinar quais espécies de valores cada variável pode armazenar. Por fim, uma variável pode ter seu conteúdo alterado a qualquer momento.

Em C, os tipos de variáveis são:

- Char: armazena letras e símbolos;
- Int: armazena números inteiros de -32767 a 32767;
- Float: armazena números entre -3.4×10^38 e 3.4×10^38 com até 6 dígitos;
- Double: armazena números entre -1.7×10^{308} até $+1.7 \times 10^{308}$ com até 10 dígitos.

A declaração de uma variável deve ser assim:

- Int idade;
- Char nomeDoAluno.

No mais, junto com a declaração, a variável também pode ser inicializada dessa forma:

- Int idade = 23.

Referências

[Nol] Cairo Noletto. *Linguagem C: o que é e quais os principais fundamentos!* URL: <https://blog.betrybe.com/linguagem-de-programacao/linguagem-c/>.

[Wik] Wikipedia. *Árvore AVL*. URL: https://pt.wikipedia.org/wiki/%C3%81rvore_AVL.

4 Desenvolvimento

Para implementação do trabalho foi idealizada a linguagem python, já para o banco de dados o MongoDB, contudo como iríamos abrigar um grande número de informações (entre 7 bilhões), preferimos utilizar a linguagem C. Pois é uma linguagem de baixo nível e seu tempo de processamento é mais rápido se comparada com uma de alto nível (python) sendo assim o banco de dados escolhido foi um arquivo txt, pois apresenta grande compatibilidade com C.

A pesquisa por “carteira de identidade” de cada país se iniciou quando deduzi, que todas as "carteiras" não teriam uma mesma nomenclatura. Inicialmente reparei que não são todos os países que possuem algo como uma carteira de identidade. Não apenas isso, como diversos países implementaram a “carteira de identidade” há pouquíssimo tempo.

A primeira identidade encontrada foi a da Alemanha, pois estava tentando fazer em ordem alfabética. Foi bem fácil de achar, pois descobri que o fluxo de brasileiros que vão morar de forma fixa na Alemanha é bem grande. Então encontrei rapidamente um site de apoio a imigrantes, que explicava como conseguir uma identidade alemã e mostrava em detalhes o que essa "identidade" possuía.

Após a Alemanha, fui procurar por alguns países da América Latina, como Argentina e Chile. Foram países bem fáceis de se achar, afinal o Brasil possui muitas parcerias com estes países, sendo até mesmo possível que um brasileiro solicite pela internet uma identidade argentina. Então de forma rápida achei

sites que ensinavam passo a passo de adquirir essas identidades e citavam detalhes sobre a "carteira de identidade".

Então o primeiro desafio foi a identidade da Bélgica, o único site que consegui achar foi o de apoio a imigração ao país, onde apenas citava o nome de sua identidade, que se chama "eID". Mas acabou revelando que os países da união europeia compartilhavam do mesmo tipo de "identidade" (com pequenas alterações de cada país) e que a Bélgica foi uma das pioneiras a assumir uma "identidade" na Europa. Após um longo tempo de pesquisa, encontrei o site oficial do governo belga, onde finalmente consegui as informações que eu precisava.

Depois da identidade belga, só foi ficando mais difíceis. Mas a experiência que tive com a Bélgica me fez aprender este novo meio: ir ao site oficial do governo. Porém a próxima "identidade" me apresentou um novo problema.

Na "identidade" da Polônia. Fui direto buscar o site do governo local, para ter informações mais precisas. Mas havia um problema, o site estava em polonês. Então a solução foi traduzir o site.

Os países seguintes eu tive que repetir o processo que fiz com o da Polônia, afinal não sei muito sobre árabe, italiano ou chinês.

Depois de conseguir uma listagem de 20 países que possuem um documento (carteira de identidade). Nesta lista alguns destes documentos já possuíam seu próprio nome, como o da Alemanha que, por exemplo, se chama "personalausweis". Mas como o da Bélgica, não possuíam o nome e era descrito apenas como "Identidade da Bélgica".

5 Exemplo de identidades encontradas

5.1 Identidade da Bélgica (eID)



Figura 6: Identidade Belgica

- Nome completo;
- Data de nascimento;
- Sexo;
- Nacionalidade;
- Número de identidade: Combinação de 12 números, no formato "xxx-xxxxxxx-xx";
- Data de emissão;
- Data de validade;

- Assinatura.

População: 11,60 Milhões de habitantes.

5.2 Identidade Argentina(DNI)



Figura 7: Identidade Argentina

- Nome completo;
 - Sexo (inclui não binário);
 - Nacionalidade;
 - Data de nascimento;
 - Data de emissão;
 - Data de vencimento;
 - Versão: Se for a primeira identidade será a “original”. Se for uma segunda via será a “B”, e assim por diante;
 - Número de documento: Contém 8 números, no esquema “xx.xxx.xxx”.
- População: 45,60 milhões de habitantes.

5.3 Identidade Israel



Figura 8: Identidade Israel

- Número de identificação: 9 números no formato “x xxxxxxx x”;
- Nome completo;
- Filiação;

- Data de nascimento;
 - Sexo;
 - Local e data de emissão;
 - Endereço atual;
 - Nome anterior (Caso possua);
 - Status (Cidadão, temporário, residência permanente);
 - Nome, data de nascimento e número de identificação da esposa e filhos (Caso possua).
- População: 9,21 milhões de habitantes.

5.4 Identidade Croácia



Figura 9: Identidade Croácia

- Nome completo;
- Sexo;
- Cidadania;
- Data de nascimento;
- Data de vencimento;
- Número de identificação: Uma sequência de 9 números no formato “xxxxxxxxx”;
- Assinatura;
- Número de documento.

População: 4,04 milhões de habitantes.

6 Conclusão

Este projeto possuía a pretensão de entender e demonstrar a estrutura de árvore AVL para fins de busca com quantidades massivas de dados. Para isso foi arquitetado um código de busca por pessoas a nível mundial, usando dos documentos de identificação única de cada país.

Com o objetivo de atingir esta demonstração de forma eficaz e obter um software de busca rápida e precisa entre centenas de milhões de dados, foram definidos pontos importantes. Primeiramente precisava ser visado como seria feito a divisão das pessoas destes países demonstrados. Com isso concluímos que o melhor método aqui para isso, era ser feito uma estrutura de árvore para cada país. Após isso, tinha que ser verificado como seria gerado todo o amontoado de dados para inserção na árvore. Essa verificação se concluiu na criação de um software a parte que seria construído unicamente

para gerar arquivos “.txt” com todos os dados pedidos. Por último foi analisado a quantidade de dados suportada, já que as árvores estariam alocando memória enquanto o programa estivesse em execução. Foi concluído que é algo que varia de hardware para hardware, com acentuação na memória RAM disponível.

E após todos os testes, a hipótese de demonstração da eficiência da estrutura de árvore AVL se confirmou como esperada. Com uma velocidade enorme na busca de dados em amontoados extremos de dados. Porém, sendo usado o método da árvore ser todo o banco de dados, é mostrado que é necessário um hardware avançado, com módulos de memória RAM de altas capacidades de armazenamento.

7 bibliografia

https://pt.wikipedia.org/wiki/Documento_de_identidade
<https://cryptoid.com.br/banco-de-noticias/o-novo-cartao-de-identidade-alemao/>
<https://manhadealemanha.wordpress.com/tag/identidade/>
<https://www.cnnbrasil.com.br/internacional/argentina-cria-documento-de-identidade-nacional-para-pessoas-nao-binarias/>
<https://www.exito-assessoria.com/post/dni-argentino>
<https://brazil.diplomatie.belgium.be/pt-br/servicos-consulares/documentos-de-viagemidentidade/carteira-de-identidade-eletronica-belga-eidkids>
<https://www.alamy.com/the-new-eid-during-pictured-during-the-launch-of-the-new-eid-electronic-identity-card-tuesday-14-january-2020-in-lokeren-belga-photo-nicolas-maeterlinck-photo-by-nicolas-maeterlinckbelgasipa-usa-image407136705.html>
<https://peopleid.zetes.com/pt/documento-de-identificacao-eletronico-belga-premiado-como-o-melhor-documento-de-identificacao>
https://pt.frwiki.wiki/wiki/Carte_d%27identit%C3%A9_%C3%A9lectronique
<https://www.chileatiende.gob.cl/fichas/3430-cedula-de-identidad-obtencion-y-renovacion>
https://stringfixer.com/pt/Resident_Identity_Card
<https://www.nbn.org.il/life-in-israel/government-services/teudat-zehut-israeli-identification/>
<https://www.cartaidentita.interno.gov.it/en/home/>
<https://www.comune.vicopisano.pi.it/Servizi/CIE-Carta-d-Identita-Elettronica>
<http://www.mup.gov.rs/wps/portal/sr/gradjani/dokumenta/licna+karta/licna+karta+>
<https://www.gov.pl/web/gov/uzyskaj-dowod-osobisty>
<https://www.comune.accadia.fg.it/index.php/posta-certificata?id=498>
<https://www.cartaidentita.interno.gov.it/en/home/>
<https://gov.hr/hr/osobna-iskaznica/350>
<https://mup.gov.hr/osobna-iskaznica-eoi/328>
<https://immichile.cl/run-rut-carnet-carne-cual-es-la-diferencia/>
<https://www.vivireenchile.cl/pt/run-dni-rut-chile-br/>

https://www.ufjf.br/jairo_souza/files/2009/12/5-Indexação-Arvore-AVL.pdf

https://pt.wikipedia.org/wiki/Árvore_AVL

<https://blog.betrybe.com/linguagem-de-programacao/linguagem-c/>