



DAVI ALVES CARVALHO - 202010711

EWERTON KEIJI ONGA - 202010161

JOSÉ VICTOR AMORIM MORAIS - 202010163

REO 07 - Relatório - Árvore B+ e Sequence Set

LAVRAS-MG

2021

Relatório - Árvore B+ e Sequence Set

1. Introdução e Metodologia

- 1.1. Este relatório consiste no resumo de todos requisitos solicitados na atividade [Implementação de Árvore B + a partir de código de Sequence Set](#), do REO 07 - Árvore B e B+, do período de 2020/2.
- 1.2. Este relatório, contém os seguintes tópicos:
 - 1.2.1. Descrição de todas estruturas utilizadas
 - 1.2.2. Descrição em alto nível da lógica do programa
 - 1.2.3. Resultado dos testes comparativos entre Sequence Set e a Árvore B+, com 20 buscas realizadas em cada estrutura
 - 1.2.3.1. Para o sorteio da busca, utilizamos um gerador de números aleatórios. Com esses números, procuramos o pokémon correspondente e fizemos a busca deste.
- 1.3. Configuração sorteada:
 - 1.3.1. Tipo de Dado: Pokémon
 - 1.3.2. Mínimo de chaves: Três (3)
 - 1.3.3. Máximo de chaves: Sete (7)
- 1.4. O arquivo **"pokedex.input"** possui as configurações de entrada utilizadas para a realização dos testes descritos neste documento. Todos os testes foram preparados e organizados para serem executados no programa implementado, de forma passível à ser compilado em um *Linux* genérico qualquer. Possíveis implementações no *Windows* podem requerer adaptações, e são passíveis de erro nos resultados.
 - 1.4.1. Para as operações de impressão, utilize o comando **'p'**
 - 1.4.2. Para as operações de depuração, utilize o comando **'d'**
- 1.5. As buscas utilizaram a função [strcmp da biblioteca "string.h"](#). Mais informações serão fornecidas no tópico 4.1.1 deste documento.

2. Descrição de todas estruturas utilizadas

- 2.1. Sequence Set (fornecido no Campus Virtual)
 - 2.1.1. *"Um Sequence Set é um arquivo binário que encontra-se dividido em pedaços de tamanhos iguais, cada uma contendo uma sequência de registros ordenados. Cada pedaço está ordenado e possui um ponteiro para o próximo pedaço (que também está ordenado em relação ao seu anterior)*
Além disso, a estrutura possui também um cabeçalho, que indica o 'pedaço' inicial do SequenceSet e a quantidade de pedaços totais."
 - 2.1.2. O sequence set implementado (previamente fornecido pelos professores) segue o conceito citado no item 2.1.1. Mais informações serão fornecidas no tópico 3.1 deste relatório.
- 2.2. Árvore B +
 - 2.2.1. *"Em uma árvore B+, os dados são armazenados nas folhas, e os nós intermediários são utilizados apenas como índices.*
Pode ser entendida como uma árvore B de índices para um sequence set. Essa é a melhor combinação, pois a árvore B permite a

busca rápida, enquanto o sequence set permite o acesso sequencial ao arquivo. “

- 2.2.2. Mais informações sobre a construção da estrutura serão fornecidas no tópico 3.2 deste documento.

3. Descrição em alto nível da lógica do programa

3.1. Sequence Set

3.1.1. Inserção

- 3.1.1.1. A inserção no Sequence Set é feita alocando uma estrutura “pacote” em um arquivo, isto é, o pacote contém um número máximo de dados e, também, um número mínimo. No caso, a quantidade previamente definida pelos professores (Tópicos 1.3.2 e 1.3.2). Enquanto o pacote não atinge seu número máximo de chaves, cada inserção é feita de maneira a ordenar os dados da página. Quando a página atinge seu número máximo de elementos e há a necessidade de se adicionar uma nova chave, uma nova página é criada de maneira que divida o número de chaves do pacote cheio ao meio. Os ponteiros que interligam as páginas são atualizados, e então o novo elemento é inserido de maneira também ordenada em uma dessas páginas.

3.1.2. Busca

- 3.1.2.1. A busca em um Sequence Set é feita de duas maneiras distintas. À priori, é realizada uma busca sequencial para encontrar a página em que o elemento buscado se encontra. Como os dados nas páginas são ordenados, se soubermos os limites de cada pacote (ou seja, o maior elemento e o menor elemento), é possível definir se o elemento se encontra em tal página ou não. Assim que a página é encontrada é feita uma busca binária para encontrar o elemento na página e retorná-lo caso encontre.

3.2. Árvore B+

3.2.1. Inserção

- 3.2.1.1. A inserção na árvore B + é feita apenas nas folhas. A estrutura da árvore (tronco) serve apenas para indexação. Dessa forma, é possível separar a inserção em 2 momentos distintos:

1) No primeiro momento, é levado em consideração que a página em que está sendo feita a inserção além de folha é a raiz (a primeira página da inserção). Assim, enquanto o número de chaves dessa página não atingir o limite pré-estabelecido, a inserção ocorre normalmente. No momento em que a página chega ao seu limite de inserções e há a necessidade de uma nova operação, uma nova página é criada, dividindo assim as chaves nessas duas páginas. O novo elemento é inserido de maneira que as duas páginas se

mantenham ordenadas como um todo. O primeiro elemento da página que contém os quatro maiores elementos é “promovido” para uma página pai, que conseqüentemente não é mais uma folha. Essa página pai, como explicado anteriormente, servirá apenas para indexação das próximas inserções ou busca.

2) O segundo momento leva em consideração que a árvore já tem páginas não folhas, ou seja, a árvore tem altura maior ou igual a um. O elemento a ser inserido é comparado com os elementos que estão no índice da árvore. Caso o elemento seja menor, ele estará em um dos filhos, caso seja maior ele será inserido entre os filhos de um dos irmãos. Isso significa que é preciso subir a árvore até chegar às folhas para a inserção ser feita do mesmo modo que ocorre no primeiro momento.

3.2.2. Busca

3.2.2.1. A busca, na árvore B+, é feita comparando o elemento buscado às chaves das páginas não folha (índices). Se o elemento buscado for menor que a chave na página, desce à árvore pelo caminho da esquerda do elemento, caso contrário, (se o elemento for maior ou igual) desce à árvore pelo caminho da direita do elemento. Seguindo dessa forma, na altura dos folhas, sempre se chega à uma página que o intervalo de suas chaves deveria conter o elemento buscado (caso ele tenha sido inserido), então é feita uma busca sequencial nesta página para encontrar o elemento.

4. Resultado dos testes comparativos das Buscas entre Sequence Set e a Árvore B+

4.1. Considerações Iniciais

4.1.1. Todas as operações de busca, tanto no SequenceSet quanto na árvore B+, foi feita usando a função [*strcmp da biblioteca “string.h”*](#). Tal função recebe 2 parâmetros, que são vetores de char. Se a função retornar um valor maior que zero, significa que o primeiro parâmetro é maior. Se o resultado for menor que 0, o primeiro parâmetro é menor que o segundo. Se, e somente se, for igual à zero, significa que os parâmetros são iguais.

4.1.2. Os resultados obtidos dizem respeito ao número de nós acessados para a operação.

4.2. Sequence Set

Busca	Nós Acessados
Hypno	12

Diglett	7
Charmeleon	6
Voltorb	22
Pidgey	18
Raticate	19
Pikachu	18
Sandslash	20
Onix	15
Krabby	13
Ponyta	18
Magnemit	14
Rapidash	18
Gengar	8
Spearow	21
Arbok	3
Golbat	10
Arcanine	4
Abra	3
Gloom	9

4.3. Árvore B+

Busca	Nós acessados
Hypno	10
Diglett	7
Charmeleon	7
Voltorb	14
Pidgey	9
Raticate	9
Pikachu	10
Sandslash	12
Onix	11
Krabby	12
Ponyta	12
Magnemit	13
Rapidash	13
Gengar	10
Spearow	13

Arbok	7
Golbat	12
Arcanine	8
Abra	5
Gloom	11

5. Conclusão

- 5.1. Com base nos resultados exibidos nos itens 4.2 e 4.3, concluímos, que, diante do banco de dados possuído, e com o conjunto de elementos sorteados para o estudo, a Árvore B+ tem vantagem de eficiência quando comparada ao Sequence Set.

Há, obviamente, casos em que o Sequence Set se sai melhor, porém são casos específicos e com menos recorrência. Enxergamos dois principais motivos para o SequenceSet acessar menos nós que a Árvore B+ em alguns casos:

1) A busca dentro da página da B+ que desenvolvemos é sequencial, enquanto a busca dentro da página do SequenceSet é binária. E, falando em eficiência, a busca binária ganha, dependendo da posição.

2) O outro caso é o elemento buscado ter um valor baixo. No caso da nossa configuração (Tópico 1.3), o “valor baixo” seriam as primeiras letras do alfabeto. Desse modo, elas estariam na primeira página do Sequence Set, e dependendo da altura da árvore, teria mais iterações até chegar nessa página.

Ao final da comparação aqui estudada, a Árvore B + possui 10,25 nós acessados, em sua média. Enquanto o Sequence Set tem uma média de 12,9 nós acessados. Isso mostra tal vantagem que o índice proporciona para a árvore. Entretanto, com esses resultados também vemos que ambas as estruturas são excelentes para a função para qual foram pensadas: Buscas rápidas e eficientes.