Relatório do Projeto II

Integrantes:

Jesus Sena Fernandes - 12697470
Tiago Chaves Bezerra - 14609637

Justificativas para Escolha de Árvores AVL para o Projeto

a. Balanceamento Rigoroso:
   As Árvores AVL mantêm um balanceamento mais rigoroso do que as Árvores Rubro-Negras. Este balanceamento garante uma altura máxima de cerca de 1,44 log(n), o que é crucial para operações eficientes de busca e leitura.

b. Altura Reduzida:
   A altura menor das Árvores AVL (em comparação com as Árvores Rubro-Negras, que podem atingir alturas de até 2 log(n)) torna operações como busca, inserção e remoção mais rápidas, pois a altura da árvore influencia diretamente o tempo dessas operações.

c. Eficiência nas Operações de Conjunto:
   Dado que o projeto envolve operações de conjunto como união e interseção, a eficiência na busca e inserção é vital. A menor altura das Árvores AVL contribui para a eficácia dessas operações.

d. Otimização da Complexidade de Operações:
   As operações de conjunto implementadas (união, interseção, inserção, remoção) têm sua complexidade diretamente influenciada pela altura da árvore. A AVL, ao garantir uma altura O(log(n)), otimiza a complexidade dessas operações.

e. Melhor Desempenho em Leitura Intensiva:
   Para aplicações com foco em leitura intensiva, como verificações de pertinência e impressão de elementos do conjunto, a AVL oferece melhor desempenho devido ao seu balanceamento e altura reduzida.

f. Rebalanceamento Eficiente:
   Apesar de as Árvores AVL necessitarem de mais rotações para manter o balanceamento em comparação com as Rubro-Negras, a eficiência geral das operações é mantida devido à altura mais baixa da árvore.

g. Adequação ao Requisito do Projeto:
   O projeto exige a implementação de uma estrutura de dados que otimize a complexidade computacional de operações específicas de conjuntos. As Árvores AVL se encaixam bem nesse requisito, oferecendo um bom equilíbrio entre eficiência de operações e complexidade computacional.

Análise da complexidade Big O das funções:

avl_cria, set_criar, item_criar, no_criar:

Complexidade: O(1)

Justificativa: essas funções realizam alocações de memória e configurações iniciais, que são operações de tempo constante.

avl_inserir, set_inserir:

Complexidade: O(log n)

Justificativa: Inserções em uma Árvore AVL exigem uma busca pela posição correta (O(log n)), seguida de possíveis rotações para manter o balanceamento.

avl_remover, set_remover:

Complexidade: O(log n)

Justificativa: similar à inserção, a remoção requer uma busca e possíveis rotações para balanceamento.

avl_busca, set_pertence:

Complexidade: O(log n)

Justificativa: a busca em uma árvore AVL é eficiente devido ao seu balanceamento, o que mantém a altura da árvore logarítmica.

avl_apagar, set_apagar, no_apagar, item_apagar:

Complexidade: O(n)

Justificativa: a liberação de memória de cada nó requer percorrer toda a árvore.

avl_imprimir:

Complexidade: O(n)

Justificativa: a impressão de todos os elementos requer percorrer todos os nós da árvore.

set_uniao, set_interseccao:

Complexidade: O(n log n)

Justificativa: essas operações envolvem percorrer duas árvores e inserir elementos na nova árvore. A inserção em uma AVL é O(log n), e isso é feito para cada elemento.

Conclusão

Portanto, a escolha da Árvore AVL para este projeto é justificada pela sua capacidade de garantir uma altura menor, o que se traduz em eficiência nas operações de conjunto. Além disso, a AVL atende aos requisitos do projeto de otimizar a complexidade computacional, sendo uma escolha mais adequada em comparação com a Árvore Rubro-Negra para este caso específico.