Introdução à Árvore Binária de Busca

Nosso trabalho apresenta um código em C++ que foca na implementação do percurso em largura em uma Árvore Binária de Busca (ABB). O percurso em largura, também conhecido como Breadth-First Search (BFS), é uma técnica que percorre a árvore nível a nível, exibindo seus elementos de forma organizada. Para realizar essa operação, o código utiliza uma estrutura de fila, que gerencia os nós da árvore a serem processados, garantindo que todos os nós de um nível sejam exibidos antes de passar para o próximo. O menu interativo permite testar essa funcionalidade, além de outras operações na ABB.

Autores: Arthur Carboni, João Pedro Nunes

Menu Interativo

Inserir Item

A opção 1 do menu permite inserir um novo item na ABB. O usuário pode inserir vários itens até digitar -1 para finalizar a inserção.

Percurso em Largura

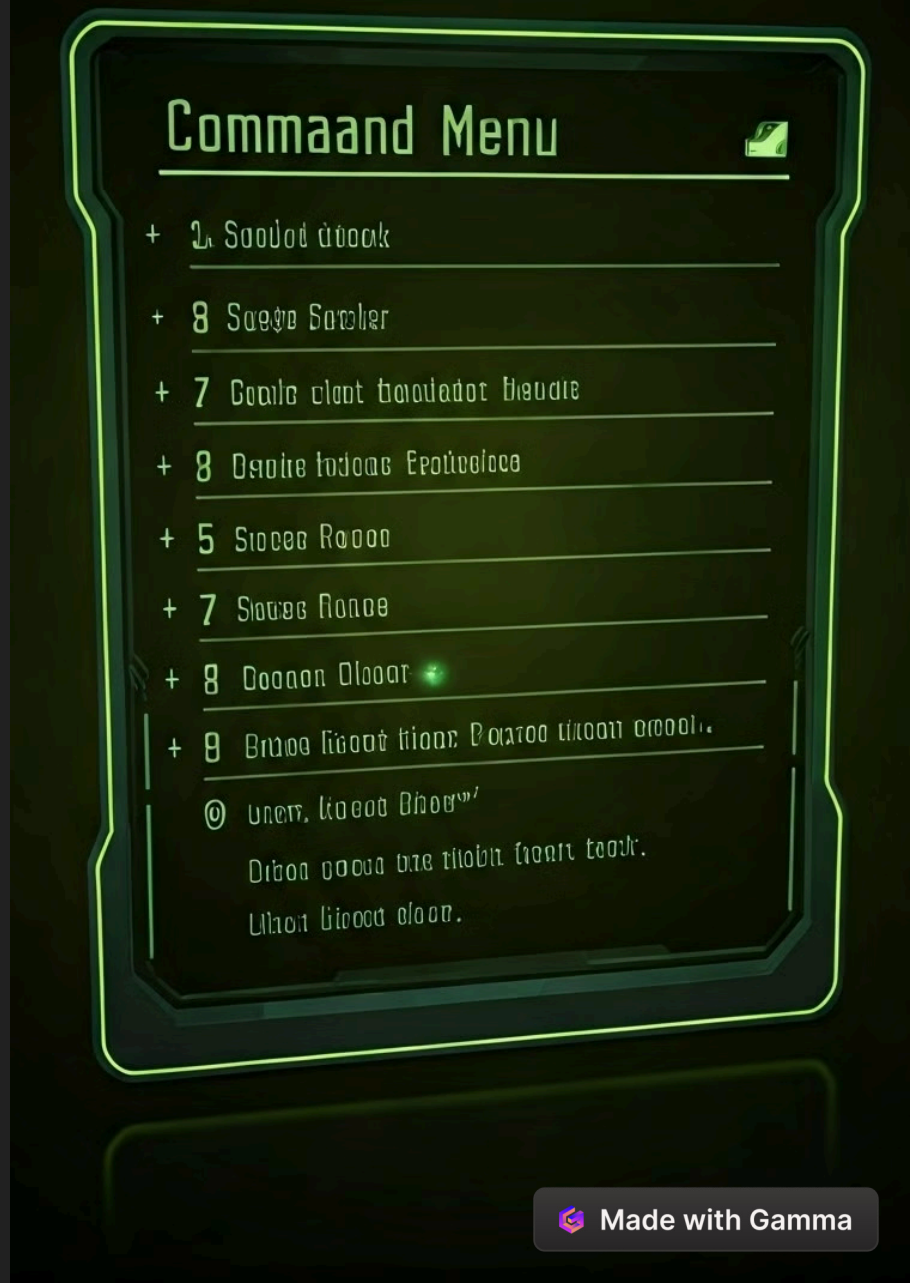
A opção 5 do menu realiza o percurso em largura da ABB, exibindo os elementos por nível. Esta opção utiliza a função percursoLargura() para percorrer a árvore.

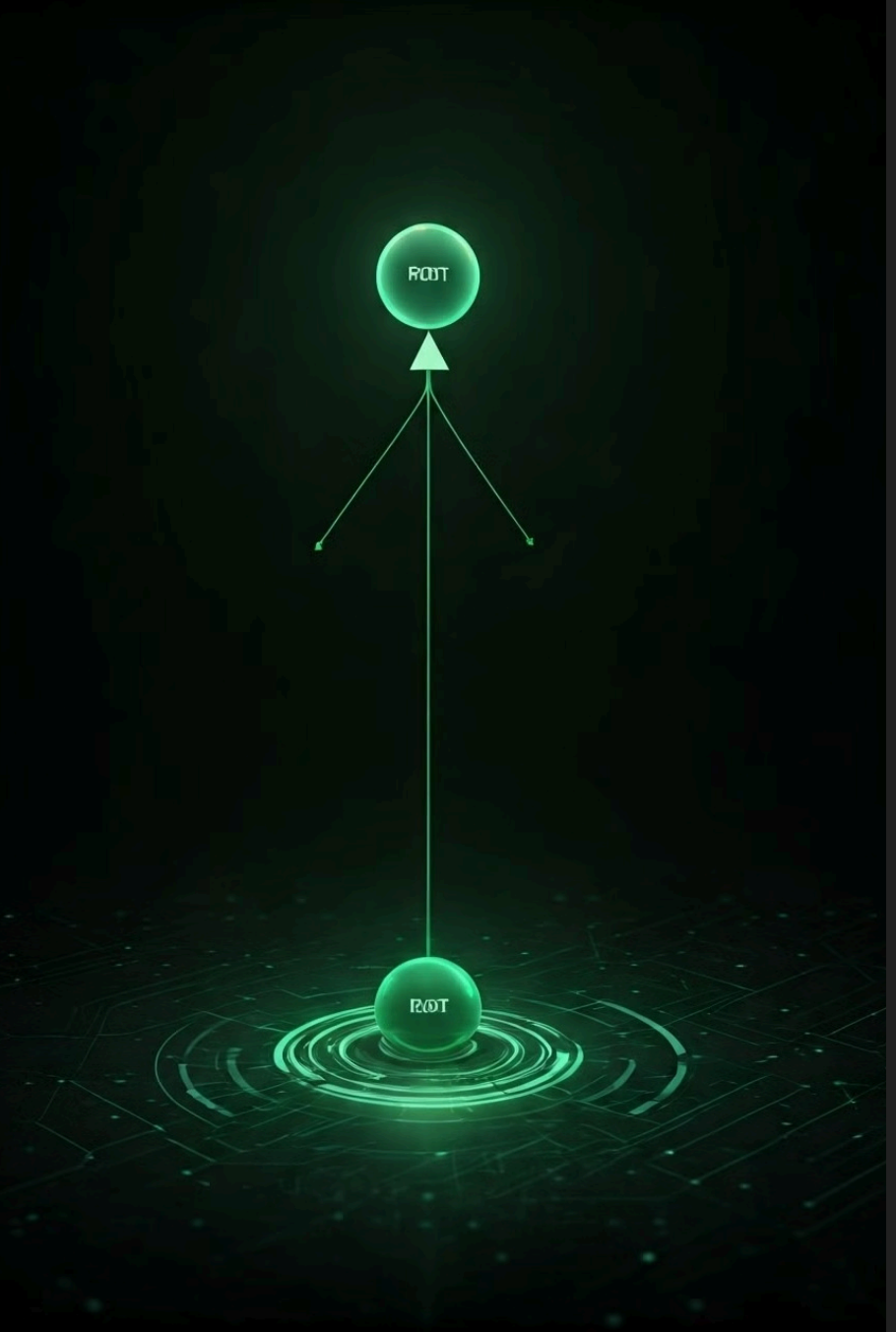
Zerar a Árvore

A opção 8 do menu libera a memória alocada para a ABB, zerando a árvore. Esta opção utiliza a função libera_AB() para liberar os nós da árvore.

Sair

A opção 9 do menu encerra a execução do programa.



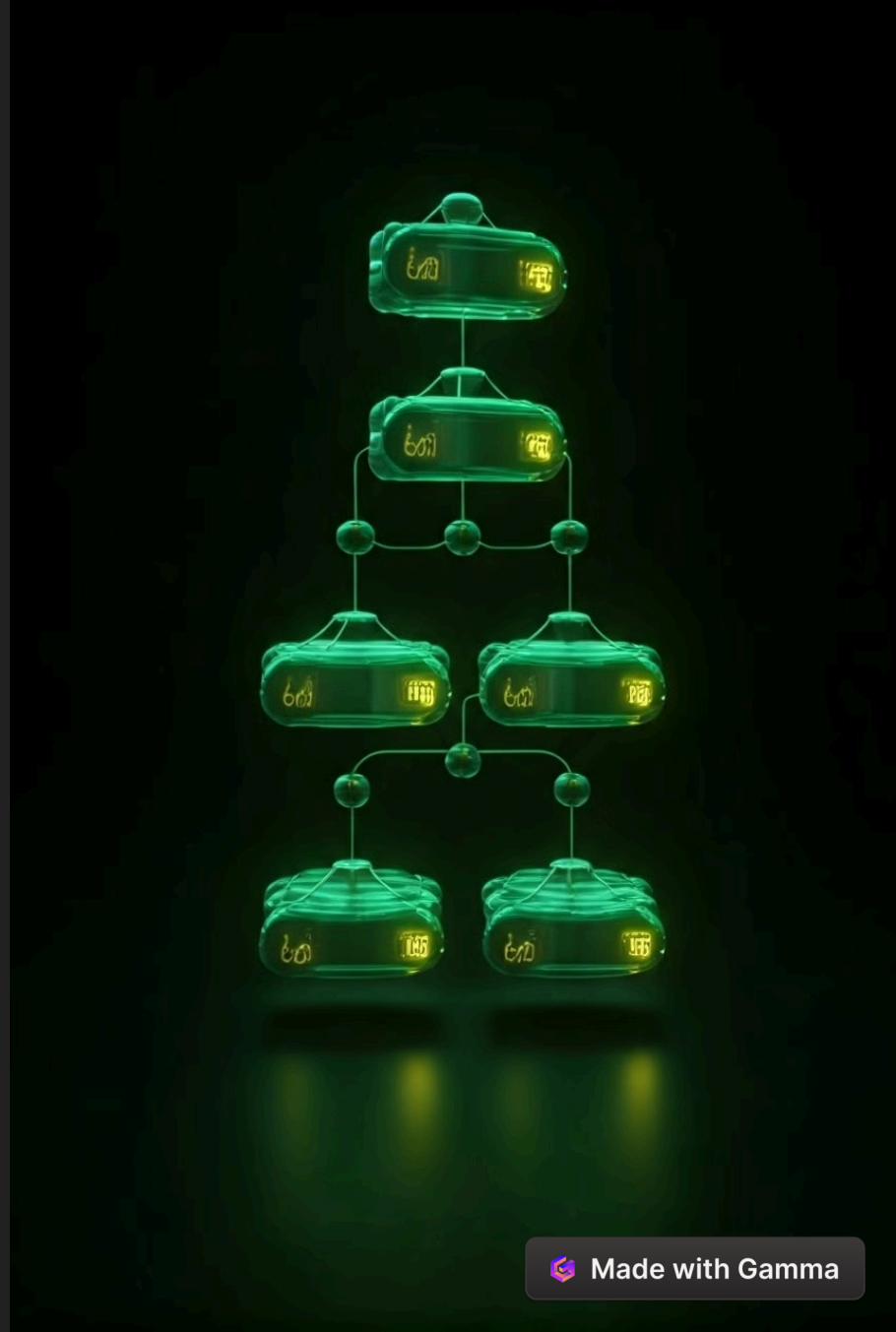


Função ini_AB()

A função ini_AB() é responsável por inicializar a ABB. Ela retorna NULL, indicando que a árvore está vazia. Esta função é chamada no início do programa para criar uma árvore vazia.

Função novoNo_AB()

A função novoNo_AB() aloca memória para um novo nó da ABB. Ela recebe como parâmetro o valor a ser armazenado no nó (x) e retorna um ponteiro para o novo nó. Se a alocação de memória falhar, a função retorna NULL.

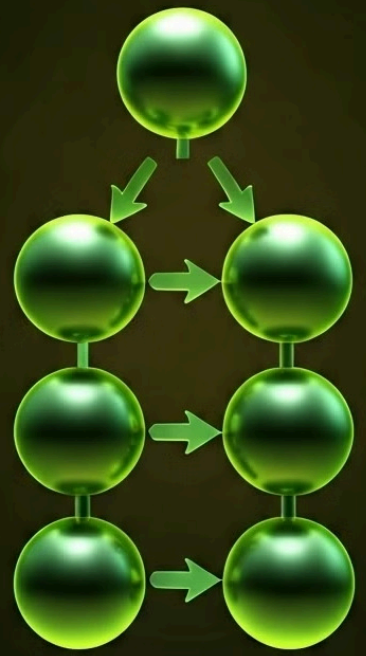


Função insere_AB()

A função insere_AB() insere um novo nó na ABB, mantendo a propriedade de ordenação da árvore. Ela recebe como parâmetros a raiz da árvore (T) e o valor a ser inserido (x). A função compara o valor a ser inserido com o valor do nó atual. Se o valor for menor, a inserção é realizada na subárvore esquerda; caso contrário, na subárvore direita. Se a árvore estiver vazia, o novo nó se torna a raiz da árvore.

Função percursoLargura()

A função percursoLargura() realiza o percurso em largura da ABB, exibindo os elementos por nível. Ela utiliza uma fila para armazenar os nós a serem visitados. A função remove o primeiro nó da fila e o visita. Em seguida, insere os filhos do nó visitado na fila, se existirem. O processo se repete até que a fila esteja vazia.

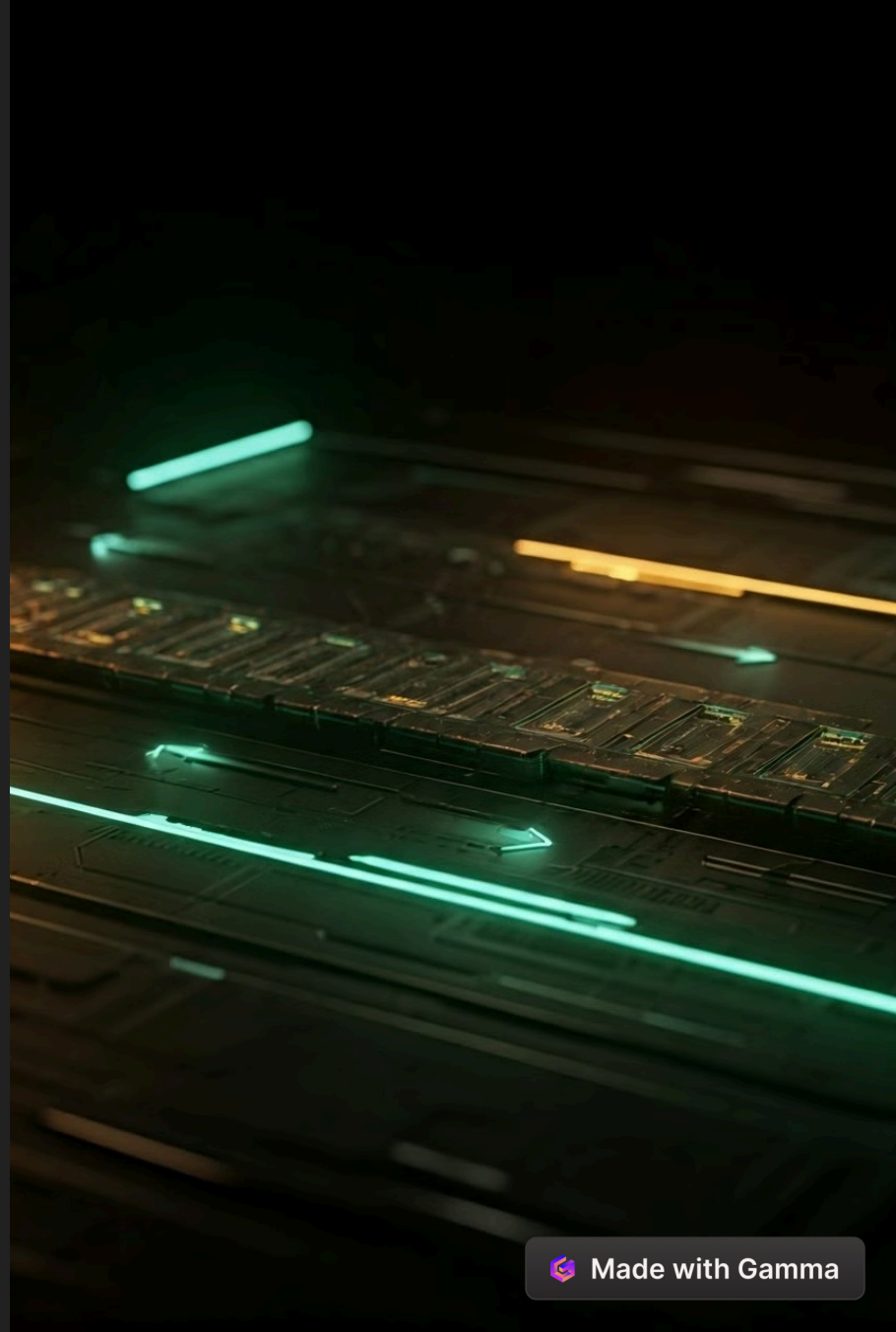


Função libera_AB()

A função libera_AB() libera a memória alocada para a ABB. Ela percorre a árvore recursivamente, liberando os nós da subárvore esquerda, depois os nós da subárvore direita e, por fim, o nó atual. Esta função é chamada para liberar a memória alocada para a árvore antes de encerrar o programa.

Fila

O código utiliza uma fila para realizar o percurso em largura da ABB. A fila é uma estrutura de dados que segue o princípio FIFO (First-In, First-Out), onde o primeiro elemento a ser inserido é o primeiro a ser removido. A fila é implementada como uma lista encadeada, com cada nó contendo um valor e um ponteiro para o próximo nó.



Conclusão

O código apresentado implementa de forma detalhada o percurso em largura em uma Árvore Binária de Busca (ABB), utilizando uma fila para gerenciar os nós e garantir a exibição dos elementos por nível. Essa abordagem facilita a visualização organizada da árvore, nível a nível, reforçando a eficiência do algoritmo Breadth-First Search (BFS). Além disso, o menu interativo permite ao usuário inserir elementos, realizar o percurso em largura e liberar a memória alocada, proporcionando uma maneira prática de testar e visualizar o comportamento da árvore.