



ESTRUTURA DE DADOS AVANÇADA – 01A – 2023.1

[Página inicial](#)[Meus cursos](#)[ESTRUTURA DE DADOS AVANÇADA – 01A – 2023.1](#)[Tópico 5. Árvore AVL](#)[Avaliação 04: árvore AVL](#)

Encerrado

Respostas

Questão 1: Dê exemplo de uma família de árvores AVL cuja exclusão de nós implica a realização de $O(\lg n)$ operações de rotação para o rebalanceamento.

Resposta: Uma família de árvores que é solução para essa questão é a família das árvores de Fibonacci T_h , que são as árvores AVL que possuem o menor número de nós para uma certa altura h fixa.

Uma árvore de Fibonacci T_h é definida recursivamente da seguinte forma:

- se $h = 0$, então $T_h = \emptyset$;
- se $h = 1$, então T_h é uma folha;
- se $h > 1$, então para a raiz r de T_h vale a seguinte regra: a subárvore esquerda de r é a árvore T_{h-2} e a subárvore direita de r é a árvore T_{h-1} .

Pela definição recursiva dada acima, obtemos imediatamente que o número de nós de T_h é dado pela seguinte relação de recorrência:

$$N(T_h) = \begin{cases} 0 & \text{se } h = 0; \\ 1 & \text{se } h = 1; \\ 1 + N(T_{h-1}) + N(T_{h-2}) & \text{se } h > 1. \end{cases}$$

O nó v que precisa ser removido da árvore de Fibonacci T_h a fim de desencadear o rebalanceamento de todos os seus ancestrais é o nó localizado mais à esquerda na árvore T_h . Quando h é ímpar, temos que v é uma folha; e quando h é par, temos que v não tem filho esquerdo mas tem uma folha como filho direito.

Ao removermos v da árvore, automaticamente o pai de v fica desregulado e passa a ter balanço igual a $+2$, o que é regulado com uma rotação simples à esquerda. A partir daí, essa situação vai se repetindo para todos os nós ancestrais de v à medida que o algoritmo recursivo de inserção vai voltando das chamadas recursivas: todos os ancestrais de v passam a ficar desregulados com balanço igual a $+2$, situação esta que é solucionada com uma rotação simples à esquerda. Isso ocorre até chegarmos à raiz de T_h , e aí com uma última rotação simples à esquerda, regulamos toda a árvore T_h . Como $altura(T_h) = O(\lg N(T_h))$, então são realizadas no máximo $O(\lg N(T_h))$ rotações.

É preciso ainda argumentar porquê os nós ancestrais vão ficando com balanço $+2$. Isso não é difícil de argumentar. Por definição de árvore de Fibonacci, todos os ancestrais p de v já tinham $balance(p) = +1$ antes da remoção. Após a remoção de v , o balanço do nó ancestral mais próximo de v salta de $+1$ para $+2$ e vai precisar ser regulado. Cada regulagem desregula o próximo ancestral mais próximo.

Questão 2: Detalhar o algoritmo de exclusão de nós em árvores AVL.

Diretrizes para a correção dessa questão: A sua tarefa aqui era pôr em palavras tudo o que eu discuti em sala e que coloquei também nos slides.

O que deve ser avaliado nessa questão:

- Foi descrito como funciona os primeiros passos do algoritmo a fim de encontrar o nó a ser deletado?
- Após encontrar o nó a ser deletado, foram discutidos os possíveis problemas que podem ocorrer ao se tentar deletar um nó que tem dois filhos, ou só um filho, ou nenhum filho? Em quais casos o aluno dividiu a análise dessa parte do algoritmo? Ele soube fazer isso?
- Após ter decidido qual nó foi removido, foram descritos os possíveis casos de remoção que podem levar à desregulagem do nó p ancestral mais próximo? Foi descrito como resolver cada desregulagem? Qual rotação foi utilizada para resolver cada caso?



Tudo isso acima precisa ter sido escrito para que a questão seja considerada completamente correta.

Questão 3. Não vou fornecer resposta para essa questão. O colega forneceu um código e você pode criar diversos casos de teste a fim de testar se a solução dele de fato funciona.

- Soluções incompletas, que não compilam, devem levar a menor nota possível

Questão 4. Essa questão foi resolvida em sala de aula, no dia 25 de abril. A solução dela está no Moodle. Para essa questão, você precisa estudar o código do seu colega e ver se realmente ele fez tudo correto, ou seja, dadas duas árvores AVL T_1 e T_2 com chaves diferentes, o código do seu colega realmente intercala as duas em uma terceira árvore AVL T_3 ?

- Códigos com falha de segmentação em alguns casos ou com erros de compilação não estão corretos.
- Atividade incompleta e que não compila deve levar a menor nota possível.
- Se tudo funcionar perfeitamente, você ainda deve checar se o tempo $O(m + n)$ foi obedecido. Discutimos em sala que a função *add* da árvore AVL não mantém a complexidade $O(m + n)$.
- Se chegar até este ponto e o seu colega tiver obedecido a todos itens acima, acredito que ele possa levar a nota máxima nesta questão. Se ele não tiver calculado os campos *height* dos nós da árvore T_3 , tudo bem, acho que isso é perdoável.

[◀ AVL Tree Visualization](#)

Seguir para...

[Descrição da estrutura de dados](#)
[Conjuntos Dinâmicos ▶](#)

