

Algoritmos e Estruturas de Dados I

2018.Q1, Matutino

Lista de Exercícios 6: Árvores Binárias de Busca

Profa. Mirtha Lina Fernández Venero

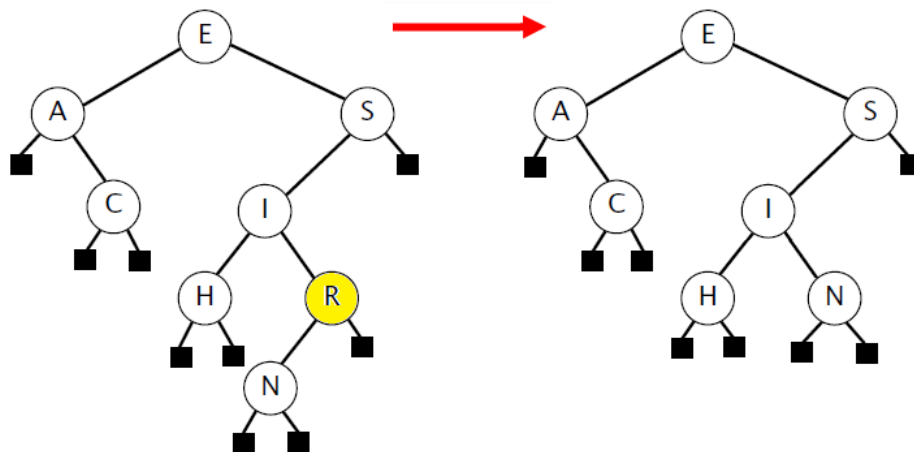
Prof. Paulo Henrique Pisani

13 de março de 2018

1 Remoção por cópia nas ABBs (T. Hibbard e D. Knuth)

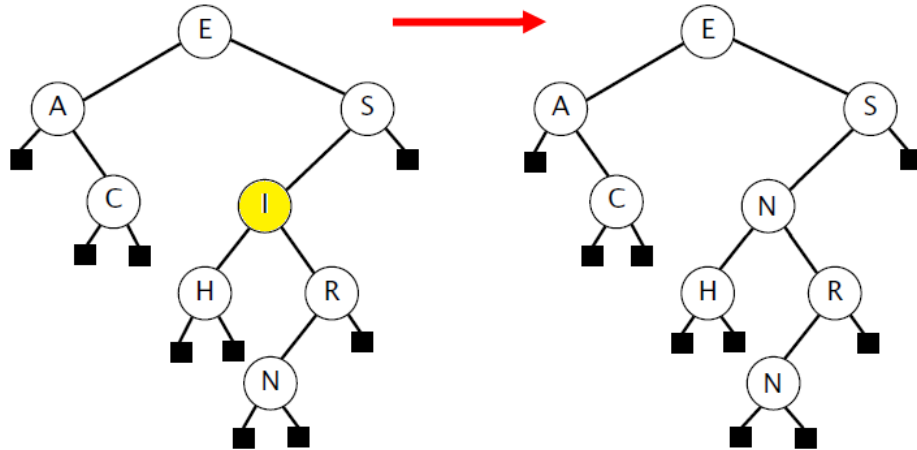
O algoritmo para remover um nó numa ABB é chamado de remoção por cópia e foi proposto por T. Hibbard em 1962. Essa remoção tem dois casos:

Caso fácil: Se o nó a remover n tem zero ou um filho $f \Rightarrow$ Atualizar o link do pai do nó com null ou f resp.

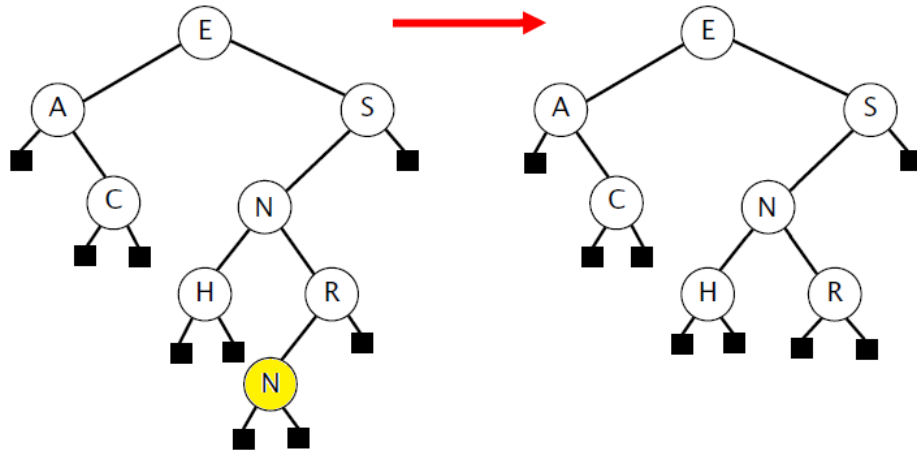


Caso difícil: Se o nó a remover n tem dois filhos \Rightarrow reduzir ao caso anterior

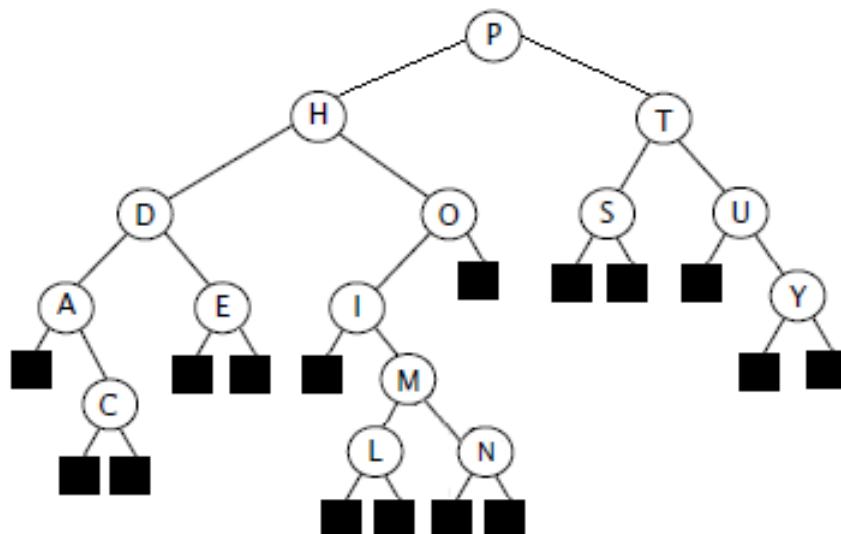
I. Colocar em n seu sucessor (menor nó à direita) $\text{succ}(n) = s = \min(n.\text{right})$



II. Remover s (note-se que S cumpre que $s.left == \text{null}$)



- Desenhe a árvore resultante de remover os seguintes nós (nessa ordem) da ABB abaixo:
D, P, A, M, T.



- Escreva uma função para remover um elemento numa ABB.

3. Escreva um programa e funções auxiliares apropriadas para testar sua implementação da função de remoção.

2 Outras operações sobre árvores binárias de busca

Implemente as seguintes operações de ordem sobre uma ABB:

1. funções para o cálculo do máximo e o mínimo.
2. funções para o cálculo do sucessor e antecessor dum elemento. As suas funções devem retornar `NULL` caso o elemento não estiver na árvore ou não existir sucessor (caso do máximo) o antecessor (caso do mínimo).

Exemplo: Na árvore do exercício 1.1 acima, o mínimo é A, o máximo é Y, o sucessor de O é P, o antecessor de I é H.

Explique qual é a complexidade das suas funções no caso pior e melhor.