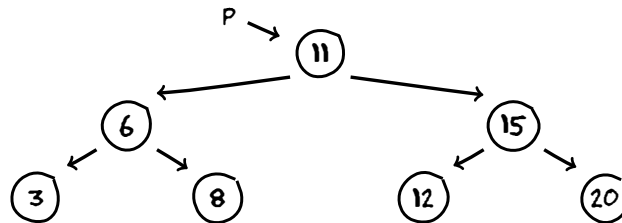


Estruturas de Dados

lista de exercícios 14

1. Imprimindo os maiores que k

Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa é

→ *imprimir todos os elementos maiores que k*

No exemplo acima, para $k = 10$, isso nos daria

=> 11, 12, 15, 20

a) Apresente um raciocínio recursivo para resolver essa tarefa..

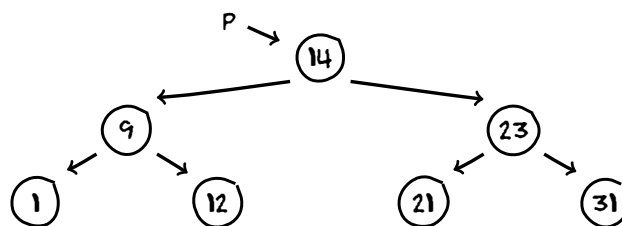
b) E apresente o código da função `main()`.

Dica: Não percorra a árvore inteira.

Passa apenas pelos elementos maiores que k — (*e aqueles que estão no seu caminho*)

2. A média dos ímpares

Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa consiste em

→ *calcular a média dos elementos ímpares da lista*

No exemplo acima, isso nos daria

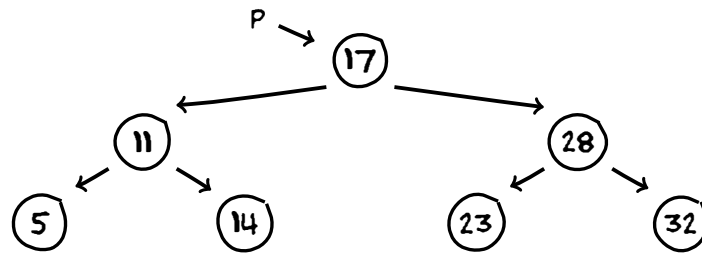
=> $85 / 5 = 17$

a) Apresente um raciocínio recursivo para resolver esse problema.

b) E apresente o código da função `main()`.

3. Imprimindo o maior ímpar

Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa é

→ *imprimir o maior elemento ímpar da árvore*

No exemplo acima, isso nos daria

=> 23

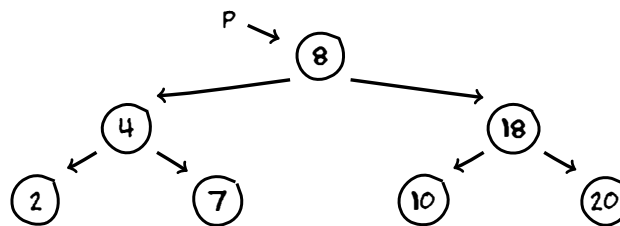
a) Apresente um raciocínio recursivo para resolver essa tarefa.

b) E apresente o código da função `main()`.

Dica: Tente percorrer a menor porção possível da árvore para encontrar o menor ímpar.

4. Encontrando o k -ésimo menor elemento

Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa é

→ *encontrar o k -ésimo menor elemento da árvore*

No exemplo acima, para $k = 5$, isso nos daria

=> 10

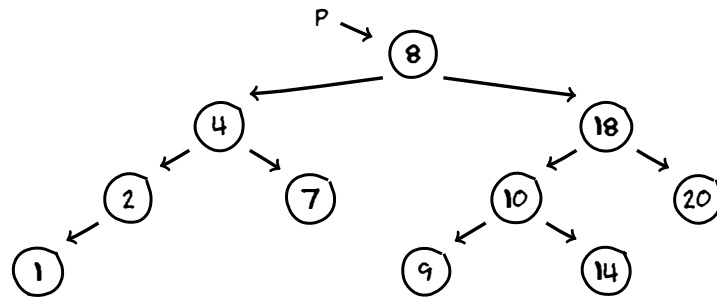
a) Apresente um raciocínio recursivo para resolver essa tarefa.

b) E apresente o código da função `main()`.

Dica: Tente percorrer a menor porção possível da árvore para encontrar o elemento.

5. Removendo todas as folhas

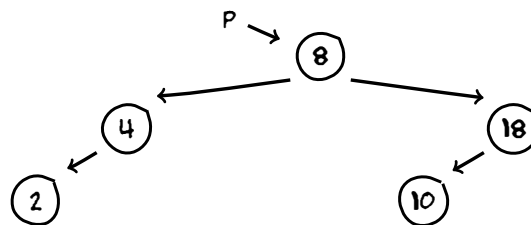
Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa consiste em

→ *remover todas as folhas da árvore*

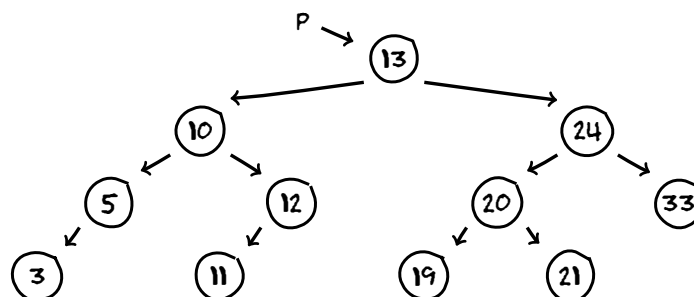
No exemplo acima, isso nos daria



- Apresente um raciocínio recursivo para resolver esse problema.
- E apresente o código da função `main()`.

6. Contando elementos balanceados

Imagine que o ponteiro p aponta para uma árvore binária de busca



Nós dizemos que um elemento é *balanceado* se as alturas das suas subárvores esquerda e direita são iguais.

A tarefa consiste em

→ *contar a quantidade de elementos balanceados na árvore*

No exemplo acima, isso nos daria

=> 3 elementos balanceados: 10, 13, 20

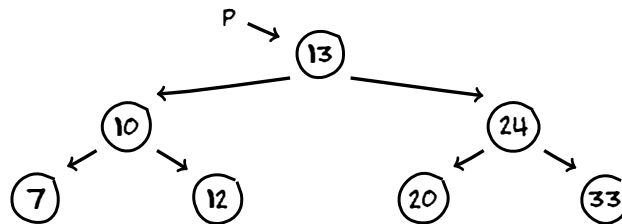
a) Apresente um raciocínio recursivo para resolver essa tarefa.

b) E apresente o código da função `main()`.

Nota: Você consegue resolver esse problema em tempo $O(n)$?

7. Duplicando o menor

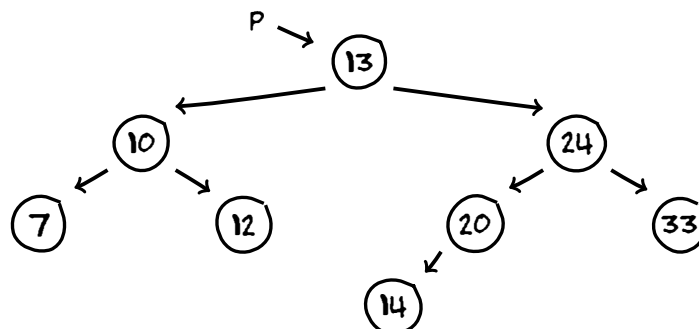
Imagine que o ponteiro `p` aponta para uma árvore binária de busca



A tarefa consiste em

→ *duplicar o valor do menor elemento da árvore, e recoloca-lo no lugar correto*

No exemplo acima, isso nos daria

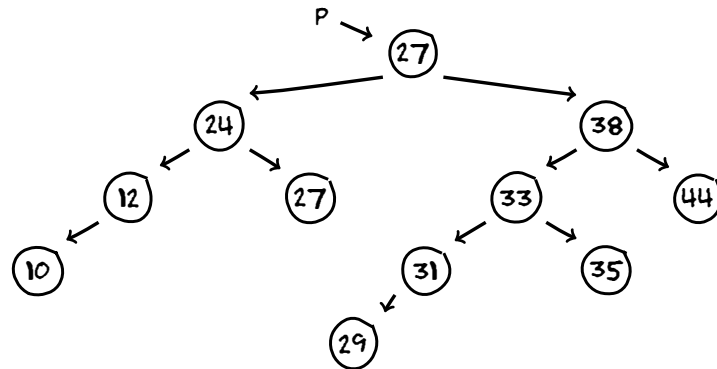


a) Apresente um raciocínio recursivo para resolver essa tarefa.

b) E apresente o código da função `main()`.

8. Desafio

Imagine que o ponteiro `p` aponta para uma árvore binária de busca elementos repetidos



A tarefa consiste em

→ *imprimir os elementos no caminho mais longo da árvore*

No exemplo acima, isso nos daria

=> 27, 38, 33, 31, 29

- Apresente um raciocínio recursivo para resolver essa tarefa.
- E apresente o código da função `main()`.