

EN 2022

A1)

(Acho que é isto que de quer, mas o código fica num loop infinito porque o j começa a 0)

A: $O(N \log N)$; B: $O(N)$; C: $O(N^2)$

A: O loop exterior acontece N vezes e o interior acontece $\log_2 N$ vezes pois o j cresce exponencialmente até N

B: O loop exterior acontece N vezes e o interior acontece um número constante de vezes, pelo que é irrelevante assintoticamente

C: A operação movelist tem complexidade $O(N)$ e acontece N vezes,

A2) - Menos tempo de computação \rightarrow Não re-calcula resultados já previamente calculados

- A profundidade na árvore recursiva mantém-se, uma vez que todas as sub-computações diferentes são feitas pelo menos uma vez.

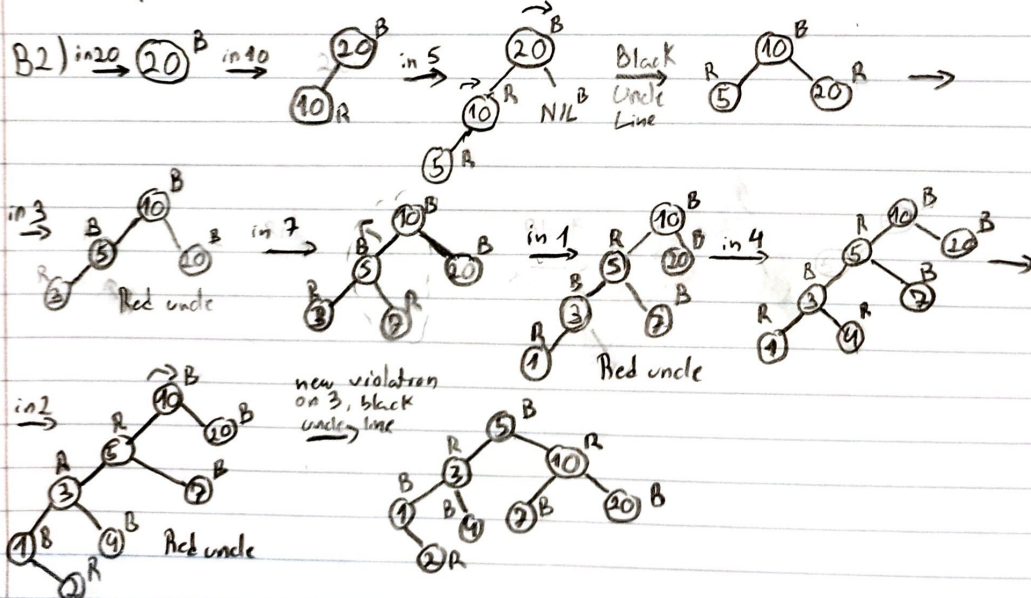
- Necessita de + memória para guardar valores memoized.

- Um programador pode usar soluções iterativas bottom-up para evitar fazer o processo recursivamente.

B1) - Fator de balanceamento tem de ser < 2 e > -2

- Altura máxima $< \lceil 1,44 \log_2(N) \rceil$

- Propriedades de uma BST normal



B3) Fator de ramificação $\equiv FT$, Node Size \sim Block Size

$$64(FT-1) + 4FT = 4096 \Leftrightarrow FT \approx 61 \quad (\text{Nenhuma das anteriores})$$

C1

Heap min

1º Ordena

2º Ordena

3º Ordena

4º Ordena

C2) B=1

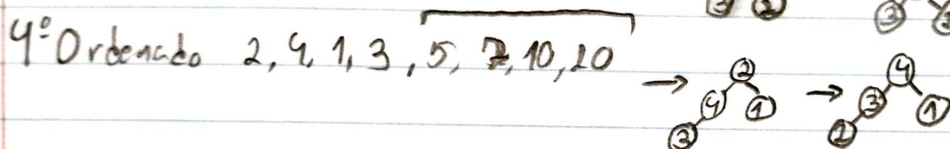
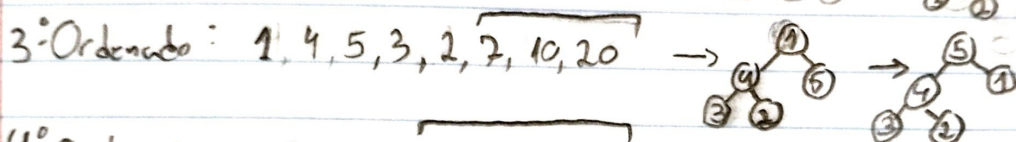
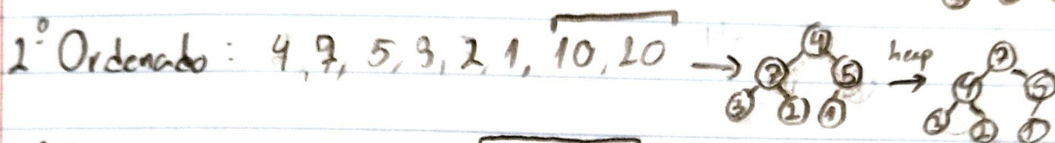
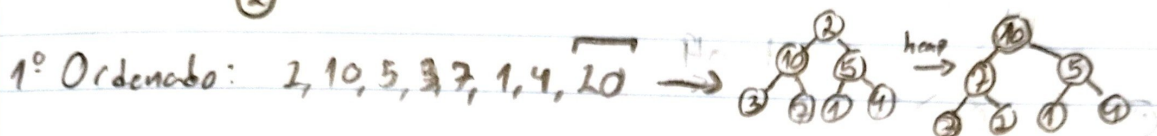
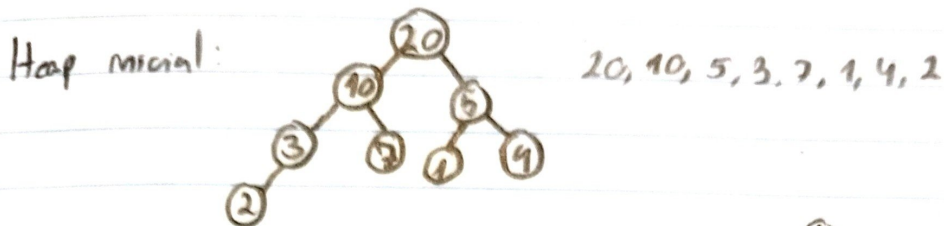
a complex

↑

NOTA: Acho que sentido, como log constante,

C3) Não

C1 20, 10, 5, 3, 7, 1, 4, 2 Arrays mostrados pre re-heap



C2) $R = \log_b(d)$ sendo b a base e d o #máximo de dígitos, portanto, caso $d = N$, a complexidade é $O(N \log N)$.

↑

NOTA: Acho que também temos de dizer que b tem de ser 2, mas isso não faz sentido, tendo em conta que \log na notação O -grande pode ser visto como \log de qualquer base, por exemplo: $\log_{10} = \frac{\log_e}{\log_e(10)}$, \log_{10} é uma constante, portanto, não pode mudar a complexidade assintoticamente.

C3) Não sei este ano