

Questão 01

A) $n + (\log n) = \theta(n)$

$$0 \leq c_1 \times n \leq n + \log n \leq c_2 \times n$$

$n = 2$ temos que:

$$0 \leq 2c_1 \leq 2 + 1 \leq 2c_2$$

$$0 \leq 2c_1 \leq 3 \leq 2c_2$$

Se $c_1 = 1$ e $c_2 = 2$:

$$0 \leq 2 \leq 3 \leq 4$$

B) $n^2 = o(n^3)$

$$\lim_{n \rightarrow +\infty} n^2 / n^3 = 0$$

C) $(n + 1)^2 = O(2n^2)$

$$0 \leq (n + 1)^2 \leq 2cn^2$$

$$0 \leq n^2 + 2n + 1 \leq 2cn^2$$

$n = 5$ temos que:

$$0 \leq 25 + 10 + 1 \leq 50c$$

$$36 \leq 50c$$

$$c > 50/36$$

D)

$$f(n) = n - 300 : \Omega(300n) \text{ e } O(300n)$$

$$\Omega(g(n)) = 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

$$0 \leq c_1 300n \leq n - 300 \leq c_2 300n$$

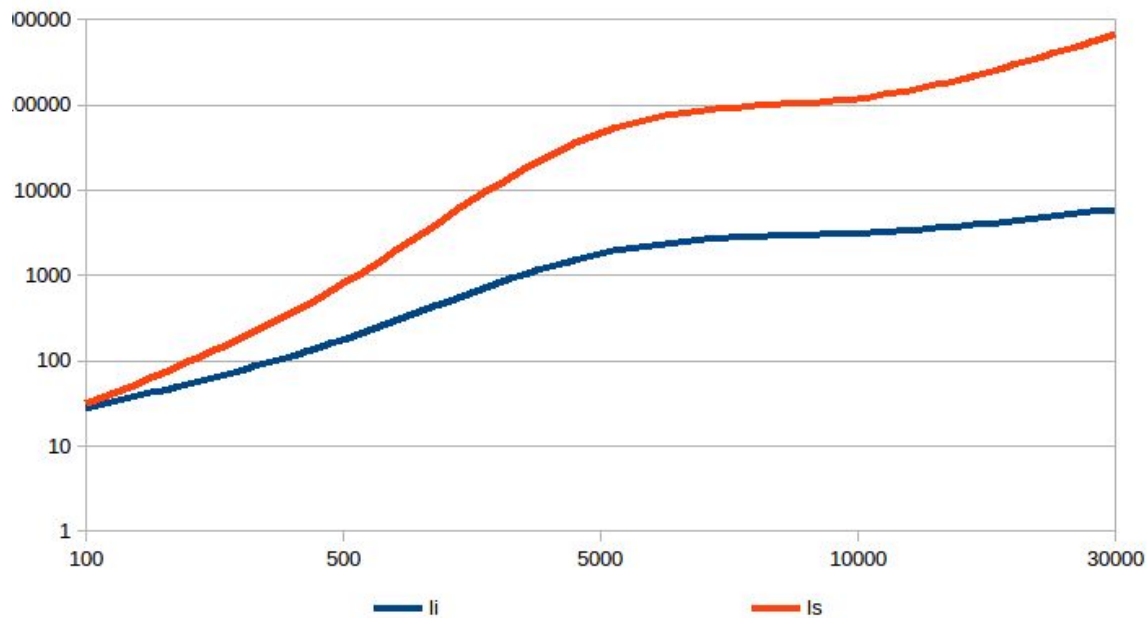
$$c_1 \geq 0 \quad n \geq 400 \quad c_2 \leq 1$$

$$0 \leq 0 \leq 100 \leq 120000$$

. :ok

Questão 02

Temos que se existe um vetor com elementos ordenáveis, então temos o problema da ordenação onde se deseja organizar esses elementos de algum jeito, dado um vetor de x_1 até x_n , para que estes dados sejam ordenados em suas posição várias repetições são feitas até o objetivo ser atingido. exemplo: caso o vetor (10, 2, 1) seja ordenado em ordem crescente é preciso que sejam realizadas n^2 comparações porque o vetor está em ordem decrescente. No caso de um vetor (1, 2, 10) é preciso que sejam realizadas apenas n comparações, onde o vetor é passeado checando se estão todos em seus respectivos lugares. Os limites superior e inferior são respectivamente $O(n^2)$ e $O(n)$.



Questão 03

A)

$$\sum_{l=1}^{10000} \sum_{i=1}^{n-5} \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 = 10000n^2 - 50000n$$

B)

$$\begin{cases} 1 & \text{se } n=1 \\ 3T\left(\frac{n}{3}\right) + \frac{5n}{3} - 2 & \text{se } n>1 \end{cases}$$

Após as recorrências se encontra:

$$3^k T\left(\frac{n}{3^k}\right) + k \frac{5n}{3} - 2 \sum_{i=0}^{k-1} 3^i$$

sendo $k = \log_3 n$, se obtém:

$$\frac{n}{2} - n \log_3 n \frac{5}{3} + \frac{3}{2}$$

C)

$$\sum_{i=1}^{n-2} \sum_{j=i-1}^n \sum_{k=1}^j 1 = \frac{n^3}{3} - \frac{3n^2}{2} + \frac{17n}{2} - 1$$

D)

$$T(n) = \begin{cases} 1 & \text{se } n=0 \\ 2T(n-1)+1 & \text{se } n>1 \end{cases}$$

Após as recorrências se encontra:

$$2^k T(n-k) + \sum_{i=0}^{n-1} 2^i$$

sendo $k=n$, se obtém:
 $2^{n+1}-1$

Questão 04

Os algoritmos de divisão e conquista consistem em dividir as tarefas grandes em tarefas menores, realizando chamadas recursivas em cada uma das partes até que o objetivo seja atingido.

```
int maior(int * v, int s, int e) {
    if (s == e)
        return v[e];
    int meio = (e - s)/2 + s ;
    int a = maior(v, s, meio);
    int b = maior(v, meio+1, e); if (a > b) return a;
    return b;
}

int menor(int * v, int s, int e) {
    if (s == e)
        return v[e];
    int meio = (e - s)/2 + s ;
    int a = menor(v, s, meio);
    int b = menor(v, meio+1, e);
    if (a > b) return b;
    return a;
}
```

Questão 05

Quick sort:

Pior caso - $O(n \log 2n)$;

Melhor caso - $O(\log n)$;

Merge sort:

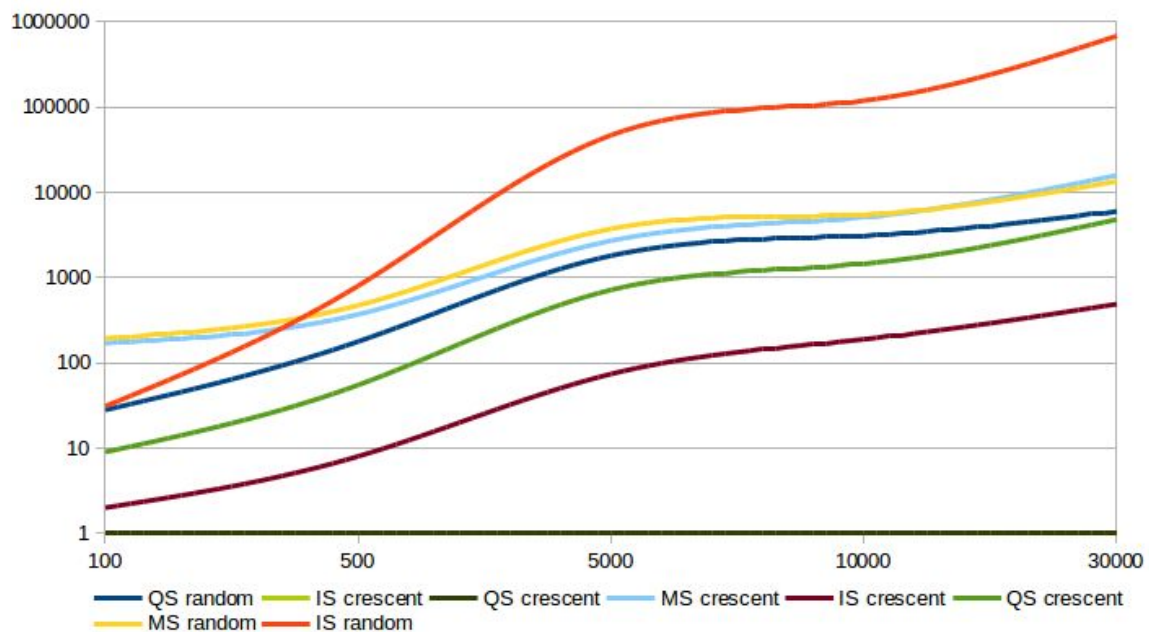
Pior caso - $O(n \log n)$;

Melhor caso - $O(n)$;

Insertion sort:

Pior caso - $O(n^2)$;

Melhor caso - $O(n)$;



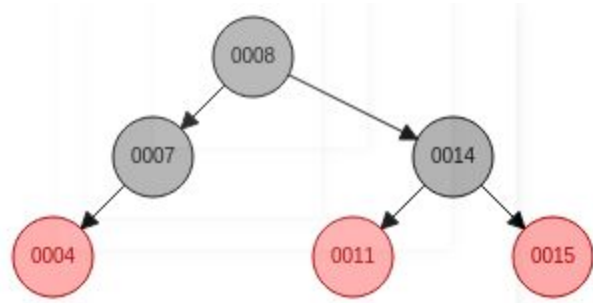
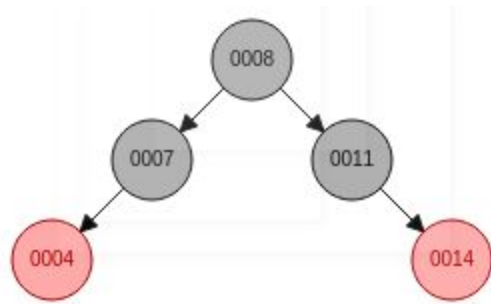
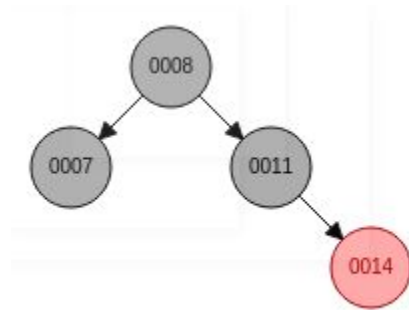
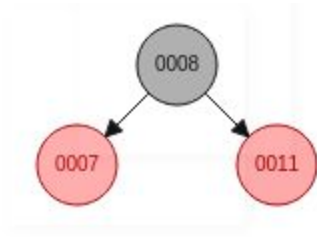
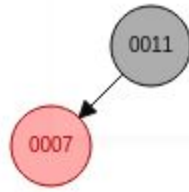
Questão 06

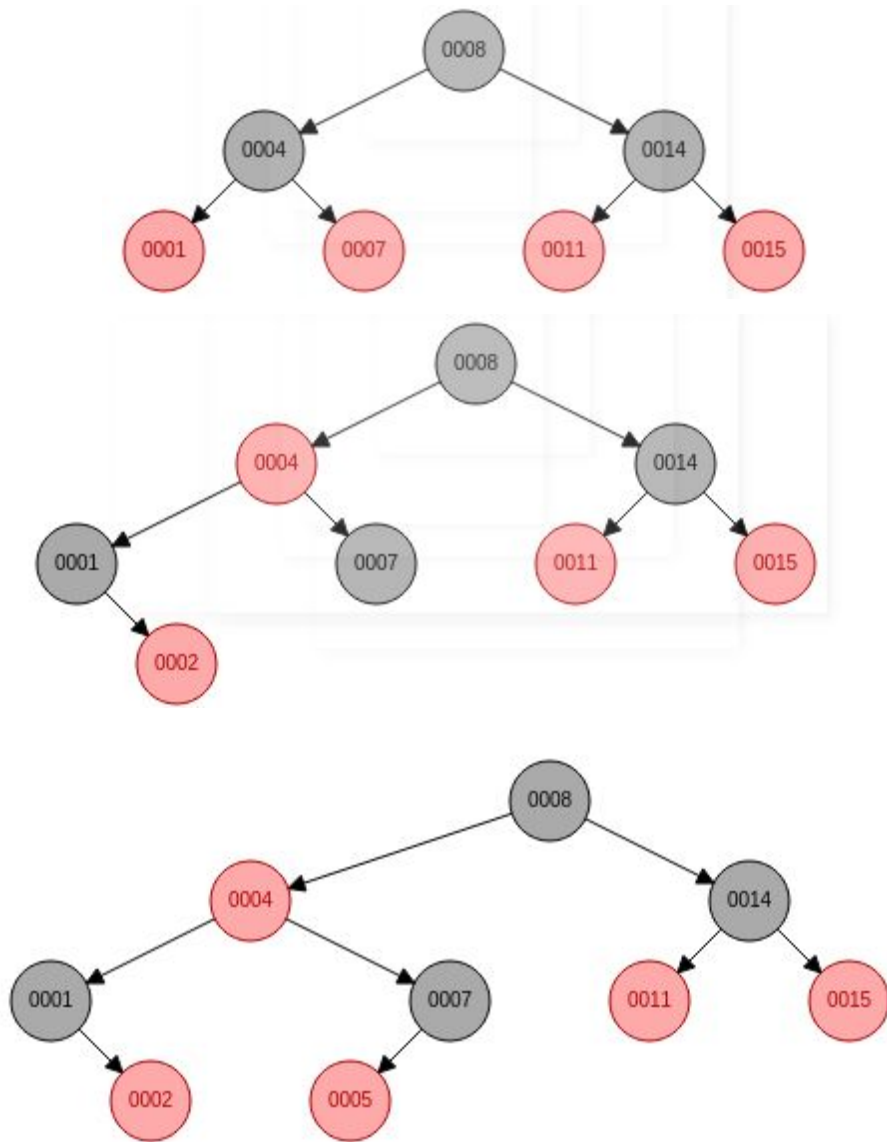
Escolhe um pivô, coloca todos os elementos menores que o pivô para à esquerda e os maiores à direita, depois o mesmo se repete até que o vetor esteja ordenado.

Questão 07

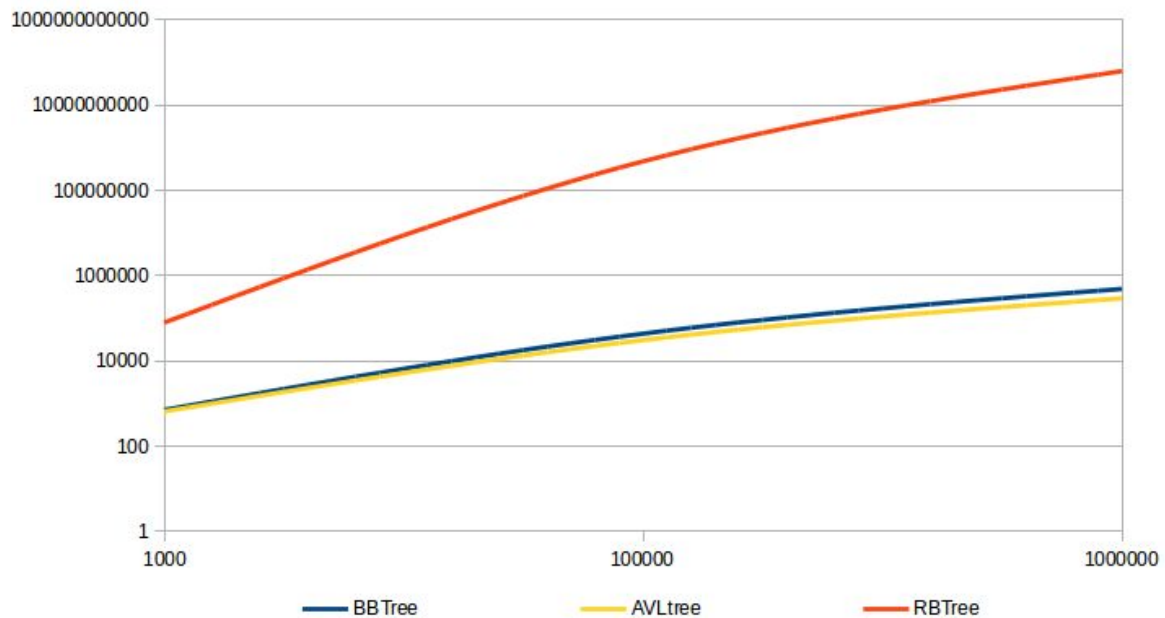
as propriedades da árvore rubro negra são:

- 1 - toda raiz é preta;
- 2 - todo filho de nó vermelho é preto;
- 3 - a quantidade de nós pretos em cada caminho da árvore é a mesma.





Questão 08



Questão 09

Séries harmônicas são séries divergentes infinitas

$$\sum_{i=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$$

O nome desta série vem do conceito de sobretons, que na música são chamados de harmônicas, são

comprimentos de onda de sobretons de uma corda vibrando que são 1/2, 1/3, 1/4...O espaçamento das notas em uma escala musical segue a série harmônica em relação aos sobretons

de cada nota, o que garante um som harmônico na escala.

Uma série harmônica é uma sequência de sons onde a frequência base de cada som é múltipla da frequência base mais baixa