**8** - A imagem mostra as complexidades de algoritmos, nos mostra a quantidade de itens processados versus o tempo utilizado para esse processamento.

Nesse gráfico temos as seguintes classificações:

**O(1) ->** algoritmos constantes, não importa o tamanho do item processado, o tempo é sempre constante sem variação;

**O(logN) ->** algoritmos com Log N tem um comportamento muito semelhante as constantes sofrem pouca variação ao longo do tempo.

**O(N) ->** algoritmos com N são considerados linear e crescem de acordo com a quantidade de itens processados.

**O(NlogN) ->** algoritmos com NlogN polinomial seu crescimento é bem próximo ao linear porém um pouco a cima.

**O(N²), O(NN) e O(N!) ->** esses algoritmos são considerados exponenciais, seu crescimento é acelerado conforma a quantidades de itens processados e são considerados intratáveis.

**9 -**

O custo final de um algoritmo pode estar relacionado a diversos fatores como:

- Tempo de execução

- Utilização de memória principal

- Utilização de disco

- Consumo de energia, de CPU, threads, etc.

Podemos medir também em outros contextos:

- Legibilidade do código

- Custo de implementação/manutenção

- Portabilidade

- Extensibilidade

- Documentação de código.

Porem calculamos o custo de execução de algoritmos dependendo do tamanho da entrada dos dados conforme imagem da pergunta anterior.

**10 –**

Códigos de loops podem apresentar uma PA ou PG.

Quando temos um loop simples (fração )é considerado O**(log N)**, já se essa PA ou PG esta dentro de um loop essa complexidade é O**(N log N)**.