

1)

- A) É quando o algoritmo não acaba perdendo tempo fazendo algo que seria redundante se o programador utiliza-se outra pegada em seu código.
- B) O número de operações que precisa ser realizado, e a quantidade de memória que existe à disposição do algoritmo, esta que recebe os dados de entrada.
- C) É utilizado o número de operações, recebendo o título de complexidade, representada pelo big O.
- D) Creio que se abstrai na questão de hardware, afinal não é algo que você consegue ter controle integral da peça.
- E)  $O(1)$ ,  $O(n)$ ,  $O(n^2)$ ,  $O(\log n)$ .

2) Ela checa a quantidade de operações realizadas, sendo principalmente utilizada para ver qual o tipo de complexidade do seu programa, ela começa a complicar quando uma variável que dita a quantidade de operações no seu código.

3)  $O(n^3)$

```
para i de 1 até n - 2 faça:
  para j de 1 até 2*n faça:
    se i for ímpar então
      para k de 1 até n2 faça:
        x(k) <- x(i) * x(j);
      fim para
    senão
      para k de 1 até n faça
        x(k) <- j;
      fim para
      x(j) <- 1;
      x(i) <- j;
    fim se
  fim para
fim para
```

Nessa parte do código possui 3 for's, que acaba têm preferência do que o linear que se encontra logo acima.