# Lista 1

1. Utilizando o Princípio de Indução Finita, prove que as igualdades abaixo são verdadeiras.

   (a) $1 + q + q^2 + \ldots + q^n = \frac{1-q^{n+1}}{1-q}$.

   (b) $\sum_{i=1}^{n} i^3 = \left[\frac{n(n+1)}{2}\right]^2$.

   (c) $n = 3a + 5b$, para $n \geq 8$ e $a$ e $b$ inteiros.

2. Considere um vetor $V$ com $n$ elementos inteiros. Considere também que uma operação de *swap* é definida como *a troca de um par de elementos do vetor em qualquer posição.* Descreva um algoritmo que, dado o vetor $V$, retorna o número *mínimo* de trocas necessárias para ordená-lo. Qual a complexidade desse algoritmo?

3. Considere um vetor $V$ com $n$ elementos inteiros. Considere também que uma operação de *swap* é definida como *remover um elemento de sua posição inicial e colocá-lo no final do vetor.* Descreva um algoritmo que, dado o vetor $V$, retorna o número *mínimo* de trocas necessárias para ordená-lo. Qual a complexidade desse algoritmo? Em qual caso especial essa complexidade se torna $O(n)$?

4. Sabemos que o algoritmo *Quicksort* possui complexidade $O(n \log n)$ no caso médio. No entanto, no pior caso a complexidade aumenta para $O(n^2)$. Descreva qual é o cenário do pior caso e para qual algoritmo clássico de ordenação o *Quicksort* "degrada" nessa situação.

5. Sabemos que para calcular o $n$-ésimo elemento da sequência de Fibonacci, podemos usar tanto uma abordagem recursiva, com complexidade $O(2^n)$, quanto uma abordagem iterativa, com complexidade $O(n)$. Entre recursão e iteração, há sempre um ganho entre uma abordagem e outra? Mais ainda, todo algoritmo que possui uma versão recursiva possui também uma equivalente iterativa?

6. As seguintes equivalências são válidas? Prove.

   (a) $2^{n+1} = O(2^n)$.

   (b) $2^{2n} = O(2^n)$.

   (c) $n^3 \log n = \Omega(n^3)$.

7. Implemente os algoritmos de busca sequencial e busca binária. Qual a complexidade de cada abordagem? Quais as vantagens de uma em relação à outra? Ambas funcionam sem restrição quanto ao vetor de entrada?

8. **(Desafio)** Resolva o seguinte problema: `http://www.urionlinejudge.com.br/repository/UOJ_1563.html`. Qual a complexidade final do seu algoritmo?