Lista de Exercícios 1:
Projeto e Análise de Algoritmos
Prof[a]. Jerusa Marchi

1. Prove, usando indução matemática, as seguintes séries:

   (a) $\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$
   (b) $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
   (c) $\sum_{i=1}^{n}(2i-1) = n^2$
   (d) $\sum_{i=0}^{n} i^3 = \frac{n^2(n+1)^2}{4}$

2. Suponha que estamos comparando implementações dos algoritmos de ordenação *insertion sort* e *merge sort* em uma mesma máquina. Para entradas de tamanho n, o insertion sort executa em $8n^2$ passos, enquanto o merge sort executa em $64n\ lg\ n$ passos. Para quais valores de n o insertion sort supera o merge sort?

3. Qual é o menor valor de n para o qual um algoritmo, cujo tempo de execução é $100n^2$, executa mais rápido que um algoritmo cujo tempo de execução é 2^n.

4. Para cada função $f(n)$ e tempo t na tabela seguinte, determine o maior tamanho n de um problema que pode ser resolvido no tempo t, assumindo que o algoritmo para resolver o problema tome tempo $f(n)$ microssegundos.

	1 seg	1 min	1 hora	1 dia	1 mês	1 ano	1 século
$lg\ n$							
$\sqrt{(n)}$							
n							
$n\ lg\ n$							
n^2							
n^3							
2^n							
$n!$							

5. Considere o problema de encontrar o máximo elemento em um vetor. Apresente o pseudocódigo para este problema, prove sua corretude e parada através do loop invariante. Apresente também as funções de complexidade obtidas pelas análises de melhor caso, caso médio e pior caso.

6. Considere a ordenação de n números armazenados em um vetor A da seguinte forma: primeiro encontre o menor elemento de A e troque-o com o elemento de $A[1]$. Então encontre o segundo menor elemento e troque-o com $A[2]$. Continue desta forma para os primeiros $n-1$ elementos de A. Escreva um pseudocódigo para este algoritmo, conhecido como *selection sort*. Qual loop invariante este algoritmo mantém? Por que ele necessita executar somente para os primeiros $n-1$ elementos ao invés de para todos os

n? Apresente o desenvolvimento das análises de complexidade do melhor caso e do pior caso para o algoritmo.