Primeiro teste de **Algoritmos e Estruturas de Dados**

15 de Outubro de 2018 **Duração máxima: 1 hora**

Justifique todas as suas respostas.

Nome:

N. Mec.:

Fórmulas:

- $\sum_{k=1}^{n} 1 = n$
- $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$
- $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2$
- $\sum_{k=1}^{n} \frac{1}{k} \approx \log n$
- $n! \approx n^n e^{-n}\sqrt{2\pi n}$

4.0 **1:** No seguinte código,

```
#include <stdio.h>

int f(int x) { return x - 2; }
int g(int x) { return x * x; }

int main(void)
{
  for(int i = -1000;i <= 1000;i++)
    if( (f(i) > 0) && (g(i) > 0) )
      printf("%d\n",i);
  return 0;
}
```

2.0 **a)** para que valores da variável `i` é avaliada a função `g(x)`?

2.0 **b)** que valores de `i` são impressos?

3.0 **2:** Ordene as seguintes funções por ordem crescente de ritmo de crescimento.

Número da função	função
1	$1.7^n + n^{1.5}$
2	$n^2 + n\log^9 n + \frac{1000}{n}$
3	$\frac{n!}{2.4^n}$
4	$n^{1.7} + 1.5^n$
5	$n\log n + n\sqrt{n}$

2.5 **3:** No seguinte código,

```
int a[10],*b = &a[7];
for(int i = 0;i < 10;i++)
  a[i] = -i;
```

qual é o valor de `b[3]`?

3.0 **4:** A complexidade computacional de muitos algoritmos é expressa usando a notação "big Oh" ($\boldsymbol{O}$) em vez da notação "Big Theta" ($\boldsymbol{\Theta}$). Porquê? (Nota: dois terços da cotação para uma boa explicação das duas notações, um terço para uma boa explicação do porquê.)

5.0 **5:** Para a seguinte função,

```
int f(int n)
{
  int i,j,k,r1,r2 = 0;

  for(i = 0;i < n;i++)
  {
    for(j = 0;j <= 4;j++)
    {
      r1 = 1;
      for(k = 0;k <= j;k++)
        r1 *= k;
    }
    r2 += r1;
  }
  return r2;
}
```

2.5 **a)** quantas vezes é executada a linha `r1 *= k;`?

2.5 **b)** que valor é devolvido pela função?

2.5 **6:** Dê um exemplo de uma função concreta que tenha uma complexidade computacional de $\boldsymbol{\Theta(n^3)}$. (Não se esqueça de justificar a sua resposta.)