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

13 de Outubro de 2017 **Duração: 1 hora**

Justifique todas as suas respostas.

Nome:

N. Mec.:

Fórmulas:

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

3.0 **1:** No seguinte código,

```
#include <stdio.h>

int f(int x) { return 2 * x + 3; }
int g(int x) { return x * x - 7; }

int main(void)
{
  for(int i = -5;i <= 5;i++)
    if( (f(i) > 0) || (g(i) > 0) )
      printf("%d\n",i);
  return 0;
}
```

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

1.5 b) que valores de `i` são impressos?

1.(a) A condição g(i)>0 é avaliada sempre que a condição f(i)>0 é falsa. Ora, temos que i assume os valores {-5,-4,...,4,5} e que a função f(x) retorna 2*x+3.
2*i+3<=0 <=> 2*i<=-3 <=> i<=-1.5 <=> i<-1, porque i é inteiro
Logo g(x) é avaliada para i quando i assume os valores {-5,...,-2}.

1.(b) Da alínea anterior, já sabemos que todos os valores de i no conjunto {-1,0,...5} são imprimidos pois, para estes valores, a condição f(i)>0 é verdadeira. Falta verificar para quais valores de i em {-5,...,-2} a condição g(i)>0 é verdadeira.
A função g(x) retorna x*x-7, logo...
i*i-7>0 <=> i^2-7>0 <=> i<=-3 V i>=3
Pelo que os valores de i impressos são {-5,-4,-3,-1,0,1,2,3,4,5}.

1.5 **2:** No seguinte código,

```
int a[10],*b = &a[7];
```

qual é o índice do elemento do array `a` que é referenciado por `b[-4]`?

(2) b[0]=a[7], b[-1]=a[6], b[-2]=a[5], b[-3]=a[4], b[-4]=a[3]
Resposta: a[3]

4.0 **3:** A complexidade computacional de muitos algoritmos é expressa usando a notação "big Oh" (O) em vez da notação "Big Theta" (Θ). Porquê? (Nota: dois terços da cotação para uma boa explicação das duas notações, um terço para uma boa explicação do porquê.)

(ver pergunta 4 do teste 2018_t1)

3.0 **4:** Ordene as seguintes funções por ordem crescente de ritmo de crescimento. Responda nesta folha, usando o número das funções na sua resposta.

Número da função	função
1	$\frac{n!}{n^{100}} - 1$
2	$n\log n + \sqrt{n}$
3	$1.2^n + 17 + n^3$
4	$23 + \frac{\log n}{n}$
5	$n^4 + \frac{1000}{n}$

relembrar:

growth rate: 1, $\log n$, $\sqrt{n}$, n, $n\log n$, n^2, n^3, 2^n, $n!$.

____ = termo dominante

Ordem: 4,2,5,3,1

Resposta:

3.0 **5:** Para a seguinte função,

(5.a)

$$N = \sum_{i=0}^{x} \left(\sum_{j=0}^{i} (1) \right) = \sum_{i=0}^{x} (i+1)$$
$$= \sum_{i=0}^{x} (i) + \sum_{i=0}^{x} (1) = 0 + \sum_{i=1}^{x} (i) + x + 1$$
$$= \frac{x(x+1)}{2} + x + 1$$

```
int f(int x)
{
  int i,j,r = 0;

  for(i = 0;i <= x;i++)
    for(j = i;j >= 0;j--)
      r += i - j;
  return r;
}
```

(5.b)

$$\sum_{i=0}^{x} \left(\sum_{j=0}^{i} (i-j) \right) = \sum_{i=0}^{x} \left(\sum_{j=0}^{i} (i) - \sum_{j=0}^{i} (j) \right)$$
$$= \sum_{i=0}^{x} \left((i+1) \cdot i - \left(0 + \sum_{j=1}^{i} (j) \right) \right)$$
$$= \sum_{i=0}^{x} \left(i(i+1) - \frac{i(i+1)}{2} \right) = \sum_{i=0}^{x} \left(\frac{i(i+1)}{2} \right)$$
$$= \frac{1}{2} \sum_{i=0}^{x} (i(i+1)) = \frac{1}{2} \sum_{i=0}^{x} (i^2 + i) = \frac{1}{2} \left(\sum_{i=0}^{x} (i^2) + \sum_{i=0}^{x} (i) \right)$$
$$= \frac{1}{2} \left(0^2 + \sum_{i=1}^{x} (i^2) + 0 + \sum_{i=1}^{x} (i) \right) = \frac{1}{2} \left(\sum_{i=1}^{x} (i^2) + \sum_{i=1}^{x} (i) \right)$$
$$= \frac{1}{2} \left(\frac{x(x+1)(2x+1)}{6} + \frac{x(x+1)}{2} \right) = \frac{x(x+1)(2x+1)}{12} + \frac{x(x+1)}{4}$$
$$= \frac{x(x+1)(2x+1) + 3x(x+1)}{12} = \frac{x^3 + 3x^2 + 2x}{6}$$

1.5 **a)** quantas vezes é executada a linha `r += i - j;`?

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

4.0 **6:** O seguinte trecho de código reserva espaço para uma matriz com $\boldsymbol{n}$ linhas com uma determinada forma. Não é reservado espaço para os elementos da matriz fora dessa forma.

```
int n;    // the number of rows of the matrix
int **a;  // the matrix

void init_a(void)
{
  int i,k,s,*p; // auxiliary variables

  // the total number of elements of the matrix
  s = ;

  // allocate memory for the array of pointers
  a = (int **)malloc((size_t)n * sizeof(int *));
  // the memory for ALL elements
  p = (int *)malloc((size_t)s * sizeof(int));
  for(i = 0;i < n;i++)
  {
    // the number of valid elements on the i-th line
    k = 2 * i + 1;
    // the pointer for the i-th line; this line uses p[0], p[1], ..., p[k-1];
    // the remaining elements of this line will never be used by a correct program
    a[i] = p - (n - 1) + i;
    // advance p
    p += k;
  }
}
```

(a)

Cada linha i, em que i pertence a {0,...,n-1}, tem 2*i+1 elementos, logo o número de elementos da matriz é dado por...

$$N = \sum_{i=0}^{n-1} (2i+1) = \sum_{i=0}^{n-1} (2i) + \sum_{i=0}^{n-1} (1) = 2 \sum_{i=0}^{n-1} (i) + n$$
$$= 2 \left(0 + \sum_{i=1}^{n-1} (i) \right) + n = 2 \sum_{i=1}^{n-1} (i) + n = 2 \cdot \frac{(n-1)n}{2} + n$$
$$= (n-1) \cdot n + n = n^2 - n + n = n^2$$

1.5 **a)** Calcule o valor a dar à variável `s` de modo a que seja alocado o número exato de elementos da matriz.

1.5 **b)** Num acesso á matriz usando `a[i][j]`, qual é a gama de valores válidos para `j`? 0...i-1

1.0 **c)** Qual é a forma da matriz?

em escada???

1.5 **7:** Dê um exemplo de uma função que tenha uma complexidade computacional de $\boldsymbol{\Theta(n^2)}$.

(7)

```
int main(int n)
{
        int counter = 0;
        int i, j;
        for (i=0; i<n; i++) {
                for (j=0; j<n; j++){
                        counter++;
                }
        }
        return counter;
}
```