Primeiro teste de Algoritmos e Estruturas de Dados

21 de Outubro de 2019 **14h10m – 15h00m**

Responda a todas as perguntas no enunciado do teste. 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 x % 3; }
int g(int x) { return x % 2; }

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

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

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

1.0 **c)** que valor de `c` é impresso?

Respostas:

3.0 **2:** Ordene as seguintes funções por ordem crescente de ritmo de crescimento. Responda nas duas colunas da direita da tabela. Na coluna da ordem, coloque o número 1 na função com o ritmo de crescimento menor (e, obviamente, coloque o número 5 na com o ritmo de crescimento maior).

Número da função	função	termo dominante	ordem
1	$\boldsymbol{n^{99} + 1.1^n}$		
2	$\boldsymbol{\frac{n!}{42^n}}$		
3	$\boldsymbol{n^2 \log n^{99} + n^2 \sqrt[3]{n}}$		
4	$\boldsymbol{1.2^n + n^2}$		
5	$\boldsymbol{\sum_{k=1}^{n}(k^2 + k)}$		

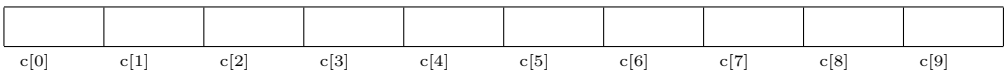
4.0 **3:** Um programador inexperiente escreveu a seguinte função para copiar uma zona de memória com `size` bytes que começa no endereço `src` para uma outra zona de memória que começa no endereço `dest`.

```
void mem_copy(char *src,char *dest,size_t size)
{
  for(size_t i = 0;i < size;i++)
    dest[i] = src[i];
}
```

Responda às seguintes perguntas, considerando que para cada uma das duas primeiras o conteúdo **inicial** do array `c` é `char c[10] = { 0,1,2,3,4,5,6,7,8,9 };`

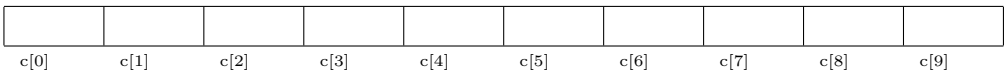
1.3 **a)** qual o conteúdo do array `c` depois de `mem_copy(&c[3],&c[5],4);` ter sido executado?

Resposta:



1.3 **b)** qual o conteúdo do array `c` depois de `mem_copy(&c[5],&c[3],4);` ter sido executado?

Resposta:



1.4 **c)** num dos casos anteriores a cópia do conteúdo de parte do array não foi feita corretamente; sugira uma maneira de corrigir este problema (não é obrigatório escrever código).

Resposta:

3.0 **4:** A notação "big Oh" é usualmente usada para descrever a complexidade computacional do pior caso de um algoritmo. Porquê?

Resposta (tente não exceder as 100 palavras):

2.0 **5:** Escreva o código de uma função que tenha uma complexidade computacional de $\boldsymbol{\Theta(\sqrt{n})}$. Como alternativa, pode optar por escrever o código de uma função de tenha uma complexidade computacional de $\boldsymbol{\Theta(\log n)}$. (Pode usar pseudo-código, se bem que uma função em `C` será mais valorizada.)

Resposta:

5.0 **6:** Para a seguinte função,

```
int f(int n)
{
  int r = -1;

  for(int i = -2;i <= n;i++)
    for(int j = i;j >= -3;j--)
      r += i - j;
  return r;
}
```

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

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

1.0 **c)** qual é a complexidade computacional da função?

Respostas: