Primeiro teste de Algoritmos e Estruturas de Dados 13 de Outubro de 2017

Duração: 1 hora **Justifique todas as suas respostas.**

**Fórmulas:**

Nome:

N. Mec.:

• Ži=n

3.0

1: No seguinte código,

**k=**

**1**

n(n+1)

#include <stdio.h>

*n(n* + 1)(2*n* + 1)

2 =

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

6

***k*=**

**1**

." 13 *(n(n*+1)

int main(void)

•

logn

1.5

1.5

1.5

for(int i = -5;i <= 5; i++)

if( (f(i) > 0) || (g(i) > 0))

printf("%d\n", i);

n! zn"e-n 2an return 0;

1.(a) A condição g()>0 é avaliada sempre que a condição fli)>0 é

falsa. Ora, temos que i assume os valores (-5,-4,...,4,5) e que a a) para que valores da variável i é avaliada a funçao g(x)? 2\*+3<=0 <=> 2+1<=-3 <=> x=-1.5 <=> ix-1, porque i é inteiro

função f(x) retorna 2\*x+3.

Logo g(x) é avaliada para i quando i assume os valores {-5,...,-2). b) que valores de i são impressos?

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 foi)>0 é verdadeira. Falta verificar para quais valores de i em (-5,...,-2} a condição g(0)>0 é verdadeira.

A função g(x) retorna x\*x-7, logo... 2: No seguinte código, 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). int a[10] , \*b = &a[7];

(2) b[0]=a[7], b[-1]=a[6], b[-2]=a[5], b[-3]=a[4], qual é o índice do elemento do array a que é referenciado por b[-4]? b[-41=a[3]

Resposta: a[3] 3: A complexidade computacional de muitos algoritmos é expressa usando a notação "big Oh” (*0*) em vez da notação "Big Theta" (O). 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) 4: Ordene as seguintes funções por ordem crescente de ritmo de crescimento. Responda nesta folha, usando o número das funções na sua resposta.

**4.0**

3.0

Número da função

função

***n!*** *n100 -1*

relembrar: growth rate: 1, log *n, n, n, n* log *n, na,* no, 2”, n!.

m log m + *Vn* 1.2" + 17+n3

log n

= termo dominante

Ordem: 4,2,5,3,1

*п*

*n4+ 1000*

***n***

**Resposta:**

Página 1 de 2 —

3.0

5: Para a seguinte função,

*1*5.a) 3.0 5: Para a seguinte função, NE Ž ( 3 (1)) - 3 (i+)

- Ž (1) + Ž (1) = 0+ Žli)+x+1

int f(int x)

int i,j,r = 0;

x (x+1)

-

++1

for(i = 0; i <= x; i++)

for(j = i;j >= 0;j--)

r += i - j; return r;

*1*5.6)

(5 1w-»)

Ź (141)-(0+ 100) = (..) - iom) Ž (1421)

1 Žlieve= ž (i+1)=(0+36)

103. (4) +0+ Ž GO) = (§ 64+ (XIX + ]*[*2X+1) + \*x+y) = xLxti)(2x1) XUSH !!

1.5 1.5

a) quantas vezes é executada a linha r += i - j;? b) que valor é devolvido pela função?

3

x(x + 1)2x+) + 3 x (x+1)

EZ

4.0

6: O seguinte trecho de código reserva espaço para uma matriz com n linhas com uma determinada forma. Não é reservado espaço para os elementos da matriz fora dessa forma.

*Ta*l

int n; int \*\*a;

*//* the number of rows of the matrix */*/ the matrix

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

void init\_a(void)

N = 2 (2+1) = 2 lail + 2 (1) = 2 2 lil +n

N

-

int i,k,s,\*p; *//* auxiliary variables

u=0

*210*

a

Ž (i)th

= 2 (n-1)n

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

= (n-1) nt n = n2ntn = n2

*//* 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;

*(*7) int main(int n)

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 a matriz usando a[i][j], qual é a gama de valores válidos para j? 0..-1 1.0 c) Qual é a forma da matriz?

em escada???

int counter = 0;

int i, j; 1.5 7: Dê um exemplo de uma função que

for (i=0; i<n; i++) {

for(j=0; j<n; j++){ tenha uma complexidade computacional de O(n).

counter++;

— Página 2 de 2 —

{

return counter;