

Nome:

nusp

Orientações

- Duração: 1h45min. Entregar a avaliação ao professor antes deste anunciar o final do tempo. Em seguida o prof. deve apontar a entrega na lista de presença. Caso o professor precise ir ao aluno recolher a avaliação será atribuída nota ZERO. Avaliações que não forem entregues receberão nota ZERO.
- Estratégia sugerida: leia todas as questões e resolva-as procurando MAXIMIZAR sua nota. Considere o tempo para ENTENDER (não só ler) o enunciado, E o que você é capaz de responder gastando pouco tempo.
- Escrever seu nome e número USP em todas as folhas. Quando houver local indicado, usá-lo.
- Preencher seu nome e número USP nesta folha.
- Entregar esta folha junto com as folhas de resposta.
- Colocar as folhas de resposta e esta uma dentro da outra de forma que formem um único bloco.
- É proibida qualquer consulta, por exemplo (não limitado a) colegas, livros, anotações feitas antes da avaliação e anotações de colegas.
- Mostrar o encadeamento lógico das idéias e conceitos é essencial nas respostas.
- As questões sobre a linguagem de programação DEVEM ser respondidas USANDO-A. Caso não haja menção sobre linguagem a usar, é permitido usar qualquer uma, inclusive pseudo-código.
- Escrever a lápis ou tinta, como preferir
- Indicar claramente a que questão refere-se a resolução
- Apresentar a resolução na ordem que preferir. (o enunciado deve ser lido sequencialmente ;-)

1. Em análise de algoritmos estudamos técnicas que nos permitem classificar algoritmos em termos do uso da capacidade de processamento. A forma mais imediata de conseguir classificar algoritmos de forma similar é medir diretamente o tempo de execução mas isto não é usual. Explique o que se mede, quais as vantagens dessa medida e qual é o processo, partindo da análise do código passando por algoritmos recursivos, fórmulas de recorrência e fórmulas fechadas, em que etapas são usadas técnicas de demonstração matemática e como isso se relaciona com notação assintótica.**2,5pt**
2. O código abaixo é compilado e executado sem erros. Simule sua execução, em especial do método fazAlgo(...). Apresente a simulação em formato de árvore de execução. A partir do código, escreva que valor corresponde ao tamanho do prob-

lema, a fórmula de **recorrência** que corresponde à quantidade de comparações que o programa executa.**2,5pt**

```
1 class Teste {
2     static void imprime (int[ ] a) {
3         for (int i=0;i<a.length;i++) {
4             System.out.print (a[i] + ",");
5         }
6         System.out.println ();
7     }
8     int[ ] metadeInicio (int[ ] a) {
9         /*Metade do inicio do array - estah correto,
10         tem que considerar a quantidade de operacoes. */
11         int[ ] r = new int[a.length/2];
12         for (int i=0;i<a.length/2;i++) {
13             r[i]=a[i];
14         }
15         return r;
16     }
17
18     int[ ] metadeFim (int[ ] a) {
19         /*Metade do fim do array - estah correto, para o
20         exemplo tem que considerar a quantidade de operacoes. */
21         int mais=(a.length % 2);
22         int[ ] r = new int[a.length/2+mais];
23         for (int i=0;i<a.length/2+mais;i++) {
24             r[i]=a[i+mais+a.length/2];
25         }
26         return r;
27     }
28
29     int[ ] fazAlgo (int[ ] a) {
30         imprime (a); // AQUI IMPRIME!
31         if (a.length<=1) return a;
32         int[ ] ini=metadeInicio(a);
33         int[ ] fim=metadeFim(a);
34         fazAlgo (ini);
35         fazAlgo (fim);
36         int i=0, iIni=0, iFim=0;
37
38         while (i<a.length) {
39             if (iIni>=ini.length) {
40                 a[i]=fim[iFim];
41                 iFim++;
42             } else if (iFim>=fim.length) {
```

```
43     a[i]=ini[iIni];
44     iIni++;
45 } else if (ini[iIni]<fim[iFim]) {
46     a[i]=ini[iIni];
47     iIni++;
48 } else {
49     a[i]=fim[iFim];
50     iFim++;
51 }
52 i++;
53 }
54 imprime (a); // AQUI IMPRIME!
55 return a;
56 }
57
58 Teste (int[ ] a) {
59     fazAlgo (a);
60 }
61
62 public static void main (String[ ] args) {
63     int[ ] a={5,3,9,7,2,1,4,8};
64     Teste t = new Teste (a);
65     imprime(a);
66 }
67 }
```

3. Apresente as definições das notações $O(g(n))$, $\Theta(g(n))$, $\Omega(g(n))$ na forma de desigualdade **E** de limite. **1,5pt**

4. Demonstre que

$$10^n \in \Theta(2^n)$$

e que

$$n^n \in \omega(2^n)$$

use a definição que preferir. Caso escolha desigualdade, é necessário demonstrar que ela é obedecida. **2,0pt**

5. Demonstre por indução (forte) que se

$$T(n) = T\left(\frac{n}{3}\right) + n$$

com $T(1) = 1$ então

$$T(n) = \frac{(3 * n) - 1}{2}$$

0,3pt para base, 1,2pt para passo