Exercícios Teóricos – u01: Fundamentos de Análise de Complexidade

Catarina F. M. Castro (803531) – AEDs II

# *Exercícios resolvidos*

1. Texto

   Descrição gerada automaticamente



1. Plote os Gráficos:  
   Texto, Carta

   Descrição gerada automaticamenteGráfico

   Descrição gerada automaticamente
2. Interface gráfica do usuário, Texto

   Descrição gerada automaticamente

Melhor caso:   
Pior caso:

1. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   O código realiza subtrações.
2. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

   Descrição gerada automaticamente  
   O código realiza
3. Interface gráfica do usuário, Texto, Aplicativo

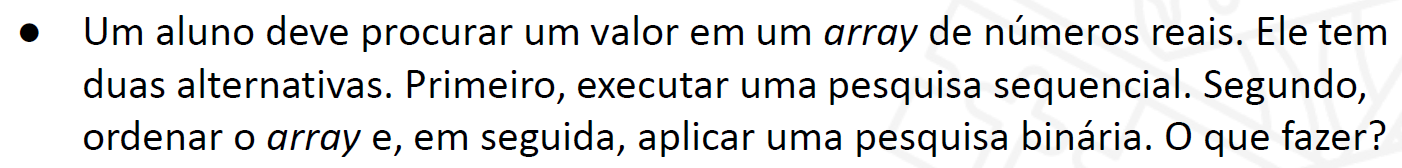
   Descrição gerada automaticamente  
   O código realiza
4. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente



A função de complexidade desse algoritmo no melhor caso é , que acontece quando o elemento buscado está na primeira posição. Já, no pior caso, a função de complexidade é , que ocorre quando se testam todas as posições, apenas para descobrir que o elemento estava na última posição ou não estava no *array*.

Visto isso, esse algoritmo é ótimo, já que é preciso testar todos os elementos para garantir a resposta.

1.   
   Nessa situação, o aluno deve optar por procurar o elemento por meio de uma pesquisa sequencial, que possui custo . Isso ocorre pois ordenar o array e depois aplicar o pesquisa binária teria o custo , e seria menos eficiente.
2. Texto

   Descrição gerada automaticamente

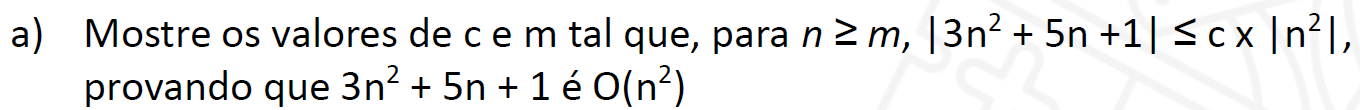
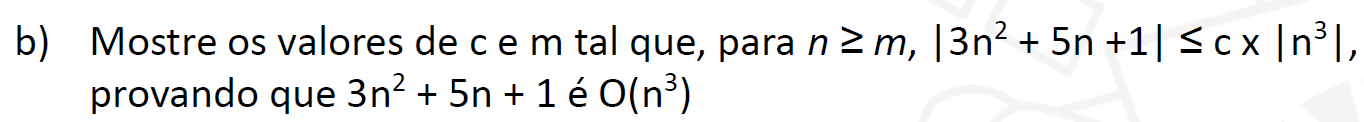


1. Interface gráfica do usuário, Aplicativo

   Descrição gerada automaticamente  
   O algoritmo de seleção faz comparações, ou seja comparações.
2. Diagrama, Texto

   Descrição gerada automaticamente com confiança média



1.   
   Para que , é preciso que . Um exemplo seria e .
2.   
   Assim como a questão anterior, para que , é preciso que , visto que a notação indica qualquer complexidade cujo crescimento seja maior ou igual ao limite justo. Um exemplo seria e .
3.   
   Não existe nenhum par em que, tal que , seja verdadeira.
4. Texto

   Descrição gerada automaticamente  
   No melhor caso, a função de complexidade é . Já no pior caso, a função é ­. Em ambos os casos, a ordem de complexidade é .
5. Texto

   Descrição gerada automaticamente  
   No melhor caso, a função de complexidade é , com ordem de complexidade , , e . Já no pior caso, a função é , com ordem, , e .
6. Interface gráfica do usuário, Texto, Aplicativo

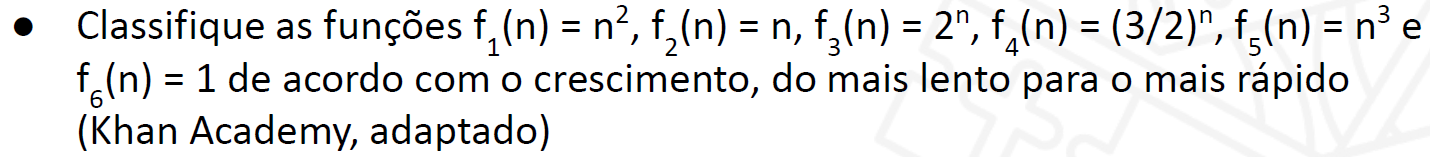
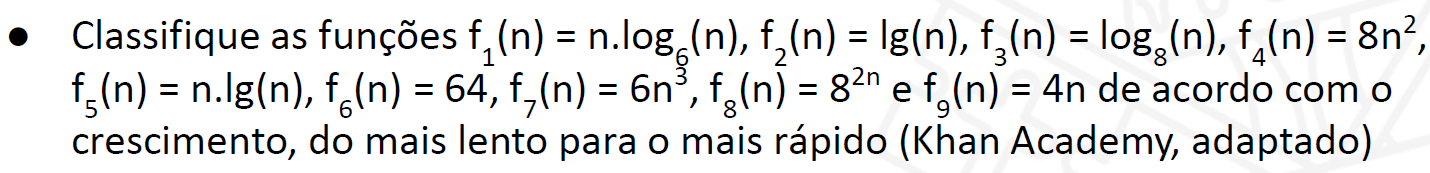
   Descrição gerada automaticamente  
   Melhor caso: , com ordem , , e .  
   Pior caso: , com ordem , , e .
7. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   Todos os casos: , com ordem , , e .
8. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   Todos os casos: , com ordem , , e .
9. Tabela, Calendário

   Descrição gerada automaticamente

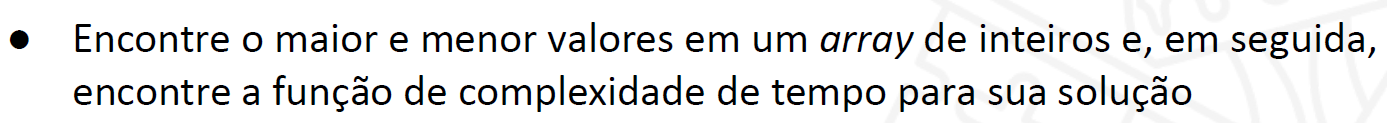


1.   
   A primeira função possui crescimento polinomial, possui crescimento linear, e possuem crescimento exponencial, possui crescimento polinomial e, por fim, possui crescimento constante.  
   Dessa forma:
2. 
3. Tabela

   Descrição gerada automaticamente



# *Exercícios não-resolvidos*



void maxMin1 () {

max = min = array[0];

for (int i = 1; i < n; i ++) {

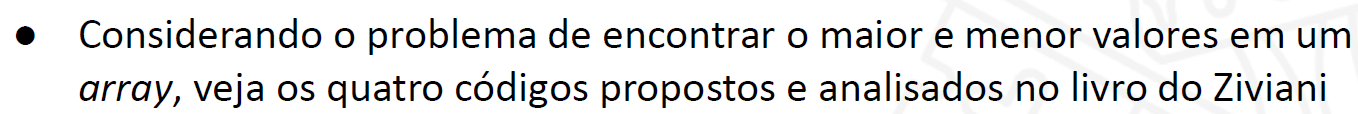
if (array[i] > max) max = array[i];

if (array[i] < min) min = array[i];

}

}

Todos os casos:

1. 
2. Tabela

   Descrição gerada automaticamente



1. Tabela

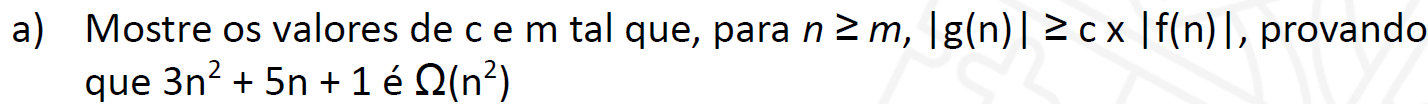
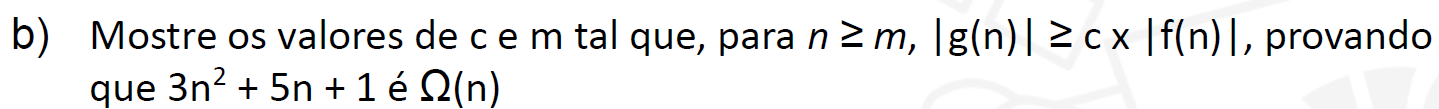
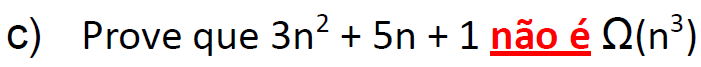
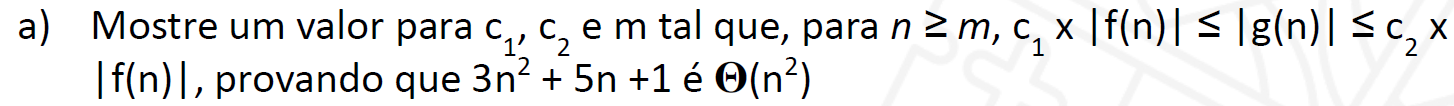
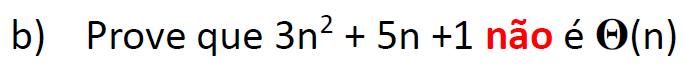
   Descrição gerada automaticamente

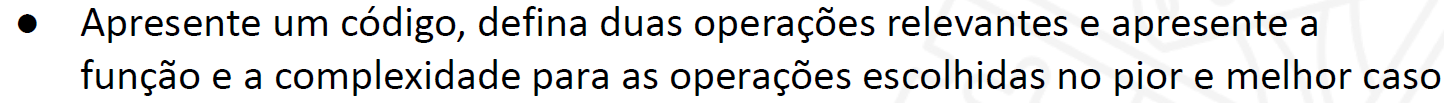


1. Tabela, Calendário

   Descrição gerada automaticamente



1.   
    e
2.   
     
    e
3.   
   Não existem valores para e que satisfaçam as condições necessárias para, tal que , seja considerado verdadeiro
4.   
    ou e .
5.   
   Não existem valores para e que satisfaçam as condições necessárias para que a afirmação seja considerada verdadeira.
6.   
   Não existem valores para e que satisfaçam as condições necessárias para que a afirmação seja considerada verdadeira.
7. Texto

   Descrição gerada automaticamente  
   No método alarme():  
   - A função de complexidade é no pior caso e é no melhor caso.  
   - A ordem de complexidade é , , e no pior caso e , , e .  
     
   Em outros métodos:  
   - A função de complexidade é no pior caso e no melhor caso.  
   - A ordem de complexidade é , , e em todos os casos.
8. 

public class ExercicioComplexidade {

public static void main(String[] args){

// Declaração do Scanner

Scanner in = new Scanner(System.in);

// Declaração de variáveis

int n = in.nextInt();  
 int arr = new vet[n];

int cont = 0;  
  
 // Preenchimento do vetor

for(int i=0, i<n, i++){

arr[i] = nextInt();

// Contagem dos números pares

if(arr[i] % 2 == 0){

cont++;

}

}

}

}  
  
No melhor caso, são feitas operações, com um nível de complexidade de . Já no pior caso, são feitas operações, também com um nível de complexidade .

1. Texto

   Descrição gerada automaticamente  
     
   As pesquisas sequenciais terão um custo de   
   As pesquisas binárias terão um custo de