UNIVERSIDADE FEDERAL DE GOIÁS – UFG
INSTITUTO DE INFORMÁTICA
SEMESTRE SELETIVO 2018/1

CURSO:	Engenharia da Computação
DISCIPLINA:	ESTRUTURA DE DADOS II
PROFESSORA:	Ma. Renata Dutra Braga (renata@inf.ufg.br ou professorarenatabraga@gmail.com)
TEMA DA AULA:	Pesquisa de dados: sequencial, binária e hashing.
AULAS	07/05/2018

ATIVIDADE
PESQUISA DE DADOS

ORIENTAÇÕES GERAIS

- Atividade em dupla.
- (Parte 1 – 02/05/2018) **Sintetize** os conceitos sobre:
  - Pesquisa sequencial
  - Pesquisa binária
  - Hashing (tabela de dispersão, hashing aberto e hashing fechado)

- (Parte 2 – 09/05/2018) **Implemente** um exemplo sobre:
  - Pesquisa sequencial
  - Pesquisa binária
  - Hashing (tabela de dispersão, hashing aberto e hashing fechado)

ENTREGA

- As atividades (parte 2 e lista de exercícios abaixo) devem ser:
  - postadas no SIGAA até o dia 13/05/2018

REFERÊNCIAS SUPLEMENTAR

VIANA, Gerardo Valdisio Rodrigues; et al. **Pesquisa e Ordenação de Dados**. Editora da Universidade Estadual do Ceará, 2015.
PEREIRA, Silvio do Lago. Estruturas de Dados em C - uma abordagem didática. 1ª edição, Editora Érica, 2016.

LISTA DE EXERCÍCIOS

Questão 01
A função a seguir implementa um algoritmo de busca binária sobre um vetor de inteiros ordenado de modo ascendente.

```
int busca(int vet[], int elem, int ini, int fim) {
      int m;

      if(fim < ini)
            return -1;

      m=(ini + fim) / 2;

      System.out.println(vet[m]);

      if(vet[m] == elem)
            return m;

      if(vet[m] > elem)
            return busca(vet, elem, ini, m-1);

      return busca(vet, elem, m+1, fim);
}
```

Essa função recebe como parâmetros um vetor (vet), o elemento que se deseja procurar no vetor (elem), o índice do primeiro elemento do vetor (ini) e o índice do último elemento do vetor (fim).

O comando System.out.println(vet[m]) exibe no console o valor do elemento de índice m do vetor vet.

Seja o seguinte vetor (vt) de inteiros:

20	25	27	38	51	57	60	65	73	74	78	80	83	88	90
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Suponha que a função busca seja chamada por meio do seguinte comando:

busca(vt, 39, 0, 14);

Qual será o 3° valor exibido no console?
a) 65
b) 51
c) 57
d) 38
e) 27

Questão 02
Considere que na Defensoria há uma lista ordenada com o nome de 1000 cidadãos amazonenses. Utilizando o método de pesquisa binária para localizar o nome de um destes cidadãos, serão necessárias, no máximo,
a) 1.000 comparações.
b) 10 comparações.
c) 500 comparações.
d) 200 comparações.
e) 5 comparações.

Questão 03
O algoritmo de busca e de ordenação que encontra o menor elemento e o troca com a primeira posição, depois o segundo menor com a segunda posição, e assim sucessivamente (n-1 vezes), usa o método de
a) seleção.
b) inserção.
c) ordenação por fusão (MergeSort).
d) ordenação por troca (BubbleSort).

Questão 04
Considere que um algoritmo de pesquisa, em um arquivo previamente ordenado, é caracterizado por realizar comparação de chaves e sucessivas divisões no espaço de busca até encontrar o termo pesquisado ou até haver um único registro. Trata-se de um algoritmo de
a) pesquisa por interpolação.
b) pesquisa binária.
c) pesquisa sequencial.
d) árvore de busca binária.

Leia os programas abaixo.

Programa 1:

```
import javax.swing.JOptionPane;
public class Busca {
    public static void main(String[] args) {
        int vet[] = {9, 21, 34, 43, 45, 60, 66, 88};
        int pri, ult, med, pos, elemproc;
        pos = 0;
        pri = 0;
        ult = 7;
        elemproc = Integer.parseInt(JOptionPane.showInputDialog("Valor?"));
        while (pri <= ult && pos == 0) {
            med = (pri + ult)/2;
            if (vet[med] == elemproc) {
                pos = med + 1;
                ult=-1;
            } else if (vet[med] > elemproc) {
                ult = med - 1;
            } else {
                pri = med + 1;
            }
        }
        if (pos==0){
           System.out.println("O valor procurado não foi encontrado");
        }else{
          System.out.println("O valor procurado é o " + (pos) + "° do vetor");
        }
    }
}
```

Programa 2:

```
import javax.swing.JOptionPane;
public class Busca {
    public static void main(String[] args) {
        int vet[] = {9, 21, 34, 43, 45, 60, 66, 88};
        int pri, ult, med, pos, elemproc;
        pos = 0;
        pri = 0;
        ult = 7;
        elemproc = Integer.parseInt(JOptionPane.showInputDialog("Valor?"));
        pos = busca(elemproc, vet, pri, ult);
        if (pos == -1) {
         System.out.println("O valor procurado não foi encontrado");
        } else {
         System.out.println("O valor procurado é o " + (pos) + "° do vetor");
        }
    }
    public static int busca(int x, int v[], int e, int d) {
        int meio = (e + d) / 2;
        if (v[meio] == x) {
            return meio + 1;
        }
        if (e >= d) {
            return -1;
        } else if (v[meio] < x) {
           return  I ;
        } else {
           return  II ;
        }
    }
}
```

Questão 05

No Programa 1, a busca pelo valor armazenado na variável elemproc

a) usa o método bubble sort para efetuar a comparação sucessiva de pares subsequentes de elementos, trocando-os de posição, se estiverem fora de ordem.

b) não apresentará resultado, pois a condição estabelecida no comando while sempre será verdadeira, gerando um laço infinito.

c) tem como base o método de seleção direta, porém, ocorrerá um erro, já que os elementos do vetor estão ordenados.

d) usa o método de pesquisa binária, normalmente mais eficiente do que o método de pesquisa linear.

e) não apresentará resultado se o elemento procurado for o 8° elemento do vetor (valor 88), já que a variável ult, que se refere ao último elemento do vetor, contém o valor 7.

Questão 06

No Programa 2, para que a busca seja realizada corretamente as lacunas I e II devem ser preenchidas, respectivamente, com
a) busca(x, v, e, meio - 1) e busca(x, v, meio + 1, d).
b) busca(v, x, meio + 1, d) e busca(v, x, meio - 1, e).
c) busca(x, v, meio - 1, d) e busca(x, v, e, meio + 1).
d) meio + 1 e meio – 1.
e) busca(x, v, meio + 1, d) e busca(x, v, e, meio - 1).

Questão 07
Avalie se são verdadeiras (V) ou falsas (F) as afirmativas a seguir.

I O método de busca "pesquisa binária" necessita de um ordenamento prévio do vetor.
II O método "pesquisa binária" possui o tempo de busca maior que o método "busca sequencial".
III O método "busca sequencial" é mais indicado quando se sabe antecipadamente que a maior parte dos registros necessita ser pesquisada.

As afirmativas I, II e III são, respectivamente:
a) V, V e F.
b) V, F e V.
c) F, V e V.
d) F, F e F.
e) V, V e V.

Questão 08
Os algoritmos de busca e ordenação são bem conhecidos no contexto da computação. Embora muita coisa tenha evoluído na área, alguns algoritmos são clássicos. Nesse sentido, de acordo com o método *algoritmoX* abaixo, é possível afirmar que ele é um algoritmo de:

```
public class Q3 {
      int algoritmoX (int chave, int vetor[], int limInferior, int
limSuperior) {
            int indice = (limInferior + limSuperior)/2;
            if (vetor[indice] == chave)
                  return indice;
            if (limInferior >= limSuperior)
                  return -1;
            else
                  if (vetor[indice] < chave)
                        return algoritmoX(chave, vetor, indice+1,
limSuperior);
                  else
                        return algoritmoX(chave, vetor,
limInferior, indice-1);
      }

      public static void main(String[] args){
            //...
      }
}
```

a) ordenação bubblesort.
b) busca sequencial ou linear.
c) busca binária.
d) ordenação quicksort.
e) ordenação heapsort.

Questão 09
Um problema de busca consiste em determinar se um dado objeto é elemento de um vetor. Sobre o algoritmo conhecido como Busca Binária, é CORRETO afirmar:
a) Sua complexidade é O(n2)
b) Quando o conjunto de dados está ordenado, sua complexidade é igual ao algoritmo de busca sequencial.
c) Não necessita de ordenação prévia do conjunto de dados. Realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com os elementos extremos do vetor. Se o elemento do início do vetor for a chave, a busca termina com sucesso. Caso contrário, se o elemento do final vier antes do elemento chave, então a busca continua até a metade posterior do vetor. E, finalmente, se o elemento do final vier depois da chave, a busca continua na metade anterior do vetor.

d) Está associado a uma estrutura de dados do tipo pilha. Se o elemento do topo da pilha for a chave, a busca termina com sucesso. Caso contrário, se o elemento do topo vier antes do elemento chave, então vão se empilhando os dados até a metade posterior da pilha. E, finalmente, se o elemento do topo vier depois da chave, vão se desempilhando os elementos da pilha.
e) É executado sobre um conjunto de dados previamente ordenado. Realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com o elemento no meio do vetor. Se o elemento do meio do vetor for a chave, a busca termina com sucesso. Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor. E, finalmente, se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.

Questão 10
Escreva uma função recursiva que receba uma cadeia de caracteres e devolva o inverso da cadeia. Por exemplo, ao receber "algoritmo" a função deve devolver "omtirogla".

Questão 11
Implemente os algoritmos de busca sequencial e binária (pesquisa e inserção) para listas simplesmente encadeadas.

Questão 12
Implemente uma remoção e uma inserção usando Hashing Aberto e Fechado.