

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL INSTITUTO DE INFORMÁTICA

Bacharelado em Ciência da Computação e Engenharia da Computação

INF 01203 - Estruturas de Dados

Aula 03 - LISTAS LINEARES - Alocação Sequencial

01. Considere os quatro trechos de código apresentados a seguir, que realizam buscas em listas lineares alocadas seqüencialmente (*arrays*).

```
#include <stdio.h>
                                       #include <stdio.h>
#include <stdlib.h>
                                       #include <stdlib.h>
#define MAX 7
                                       #define MAX 7
                                       int busca2(int vet[], int n)
int buscal (int vet[], int n)
                                         int i = 0;
 int i = 0;
                                         vet[MAX] = n;
  while(i < MAX){</pre>
                                         while(vet[i]!=n)
     if (vet[i] == n)
                                              i++;
                                         if (i == MAX)
       return i+1;
     else
                                           return -1;
       i++;
                                         else
                                            return i+1;
  return -1;
                                       int main()
int main()
                                         int v[]={2,4,5,1,6,9,3,0};
  int v[]={2,4,5,1,6,9,3};
                                         int resultado;
 int resultado;
                                         resultado = busca2(v, ???);
 resultado = busca1(v, ???);
                                         printf("%d\n", resultado);
 printf("%d\n", resultado);
                                         system("pause");
  system("pause");
#include <stdio.h>
                                       #include <stdio.h>
#include <stdlib.h>
                                       #include <stdlib.h>
                                       #define MAX 7
#define MAX 7
int busca3(int vet[], int n)
                                       int busca4(int vet[], int n)
 int i = 0;
                                         int inf = 0, sup = MAX -1, meio;
  vet[MAX] = n;
                                         while(inf <= sup) {</pre>
 while(vet[i]<n)</pre>
                                            meio = (inf + sup)/2;
       i++;
                                            if (vet[meio] == n) {
  if ((i == MAX) || (vet[i]!= n))
                                                  inf = sup + 1;
    return -1;
                                                  return meio;
                                            else if (vet[meio] < n) inf = meio +1;</pre>
     return i+1;
                                                  else sup = meio - 1;
int main()
                                        if (inf >= sup) return -1;
  int v[]={1,2,4,5,6,7,8,0};
  int resultado;
                                       int main()
 resultado = busca3(v, ???);
 printf("%d\n", resultado);
                                         int v[]={1,2,4,5,6,7,8};
  system("pause");
                                         int resultado;
}
                                         resultado = busca4(v, ???);
                                         printf("%d\n", resultado);
                                         system("pause"); }
```

1 – O que faz o trecho de código busca1()?
2 – O trecho de código busca1 () é indicado para (marque X na(s) alternativa(s) correta(s)) :
() lista ordenada (ordem crescente)
() lista ordenada (ordem decrescente)
() lista desordenada 3 – O que faz o trecho de código busca2 ()?
5 – O que laz o treello de codigo buscazo.
4 – O trecho de código busca2 () é indicado para (marque X na(s) alternativa(s) correta(s)) :
() lista ordenada (ordem crescente)
() lista ordenada (ordem decrescente)
() lista desordenada
5 – Considerando as funções busca1() e busca2(), qual delas é mais "eficiente"
considerando o número de comparações executadas? Justifique sucintamente sua resposta.

7 – O trecho de código busca3 () é indicado para (marque X na(s) alternativa(s) correta(s)) :
() lista ordenada (ordem crescente)() lista ordenada (ordem decrescente)
() lista desordenada
8 – O que faz o trecho de código busca4 ()?
9 – O trecho de código busca4 () é indicado para (marque X na(s) alternativa(s) correta(s)) :
() lista ordenada (ordem crescente)
() lista ordenada (ordem decrescente)
() lista desordenada
10 – Considerando as funções busca3() e busca4(), qual delas é mais "eficiente" considerando o número de comparações executadas? Justifique sucintamente sua resposta.