

**Lista Complementar – Algoritmo de Busca**  
**IC 2**  
**Unesp – Rio Claro- Prof. Rafael Oliveira**

---

**1-)** Baixe o programa “busca\_binaria.c” e execute o código em sua máquina.

**2-)** Analise o algoritmo abaixo e explique qual tipo de busca é implementada:

```
int Busca ( int k[], int chave , int N)
{
    int inf,sup,X;
    inf=0;
    sup=N-1;
    while (inf<=sup)
    {
        X=(inf+sup)/2;
        if (chave==k[X])
            return X;
        else if (chave<k[X])
            sup=X-1;
        else
            inf=X+1;
    }
    return -1;
}
```

**3-)** Implemente os algoritmos de busca sequencial e binária (pesquisa e inserção) para listas simplesmente encadeadas.

**4-)** O seguinte algoritmo de busca sobre um vetor ordenado é conhecido como busca de Fibonacci devido ao uso de números de Fibonacci.

```
for (j=1;fib(j) < n;j++); /* inicializando j */
mid = n – fib(j-2) + 1;
f1 = fib(j-2)
f2 = fib(j-3);
while(key != k(mid) )
if ( (mid < 0) && (key > k(mid))
{
    if (f1 == 1)
        return -1;
    mid += f2;
    f1 -= f2;
    f2 -= f1;
}
else {
```

```
if (f2 == 0)
return -1;
mid -= f2;
aux = f1 - f2;
f1 = f2;
f2 = aux;
}
return mid;
```

Explique como esse algoritmo funciona. Compare a quantidade de comparações de chave realizada por este algoritmo com o de busca binária. Modifique a parte inicial desse algoritmo para que ele calcule os números de Fibonacci eficientemente, em vez de consultar seus valores em uma tabela ou calcular cada um de novo.

**5-)** Quais as propriedades de uma árvore binária de busca?

**6-)** O que é uma árvore binária balanceada?

**7-)** Leia os slides da aula 19-1 (conceitos de árvores). Realize os exercícios conceituais apresentados no final da aula.