

PESQUISA EM VETORES		PESQUISA EM VETORES
O que é uma <b>Operação de Pesquisa</b> ?		<b>Operação de Pesquisa:</b> podemos exemplificar uma operação de pesquisa com um programa de controle de estoque que é capaz de buscar um determinado dado, um código numérico ou um nome, a descrição e as características de um determinado produto. Os algoritmos de busca (pesquisa) permitem encontrar um determinado dado no meio de um milhão de dados. Entretanto, por mais rápidos que sejam os computadores hoje, esse processo pode ser bastante demorado.
O que é um <b>Algoritmo de Pesquisa Binária</b> ?		Os Algoritmos de Pesquisa Binária são algoritmos que devem ser usados sempre em vetores estiverem ordenados. O algoritmo de busca binária tem como idéia básica o fato de que, dada uma posição dentro do vetor, o elemento procurado vai estar, ou antes, ou depois desta posição, e, portanto, não se precisa procurar dos dois lados. Numa dada lista ordenada: 1º - Compara-se o elemento procurado com o registro do meio da lista, para saber se o elemento estaria na metade superior ou inferior da lista; 2º Repete-se esse processo até encontrar o elemento ou esgotar a lista;
Ilustre como poderíamos criar um Algoritmo de Pesquisa Binária...		A função a seguir contém um Algoritmo de Pesquisa Binária em um vetor que tem o objetivo de encontrar o valor índice no vetor para um determinado elemento procurado, se o elemento existir dentro do vetor, esse algoritmo retorna o valor índice dele, mas, se ele não existir o valor de retornado é "-1". Perceba que primeiro ele procura o elemento no meio do vetor, se não encontra ele vai estendendo a busca para as extremidades do começo e do final do vetor, até encontrar o valor....
- reservada para a questão acima -		<pre>int busca_binaria (int vet[], int tam, int elem){      int comeco, fim, meio, pos;     pos = -1;     comeco = 0;     fim = tam - 1;      while (comeco &lt;= fim){         meio = (int) ( (comeco + fim) / 2 );         if (elem == vet[meio]){    //elemento foi encontrado             pos = meio;             comeco = fim + 1;         }         else if (elem &gt; vet[meio])    //elemento pode estar na parte direita do vetor             comeco = meio + 1;         else fim = meio - 1;    //elemento pode estar na parte esquerda do vetor         }         return(pos);     } }</pre>