UniAcademia

Disciplina

Estrutura de Dados

prof. Jacimar Tavares
jacimar.tavares@gmail.com

Módulo 05

Pesquisa

Pesquisa

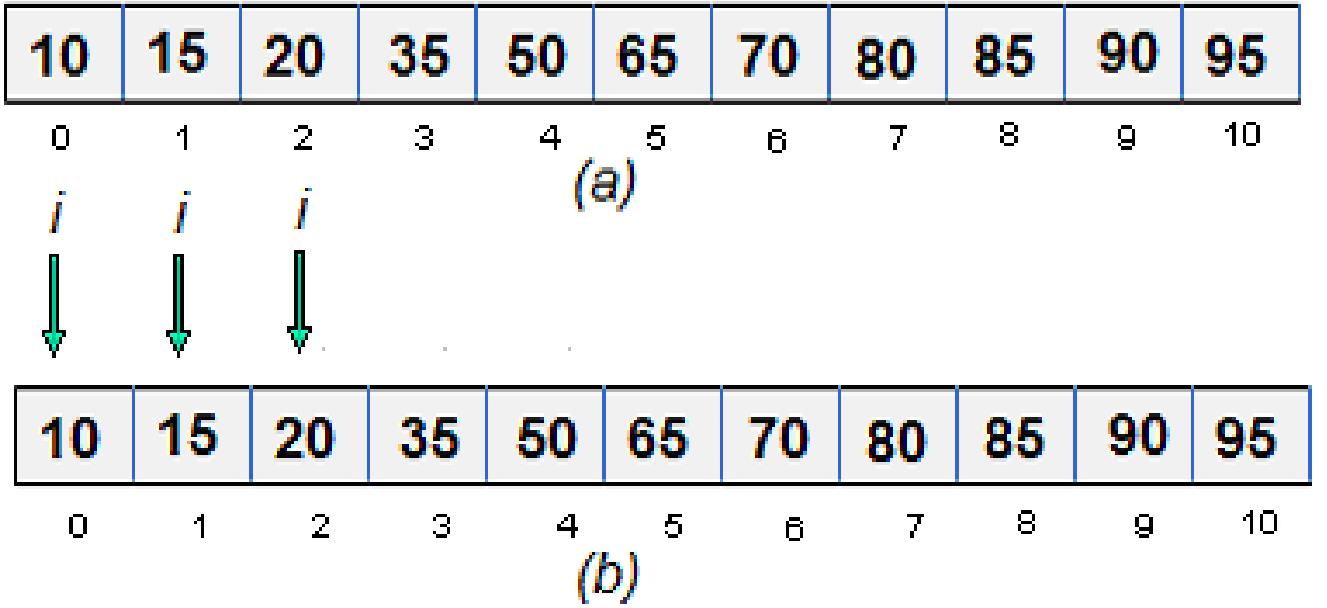
- **Overview**
  - Pesquisa é a forma como encontramos elementos em um conjunto de dados.
  - Exemplos?
    - Busca em conjunto de dados de produtos
    - Busca em arquivos.
    - Busca em estruturas de dados..
    - etc

Pesquisa

- **Overview**
  - Varredura: Pesquisa sequencial
    - É a forma mais simples e primitiva de pesquisa em um arquivo sequencial.
    - Este procedimento consiste em varrer o arquivo até o fim e só parar quando
      - achar a chave, ou
      - quando a chave de pesquisa for maior que a próxima chave do arquivo ou
      - ainda quando for fim de arquivo.
    - A varredura é uma técnica que consiste basicamente de se varrer o arquivo sequencialmente do início até o seu final.

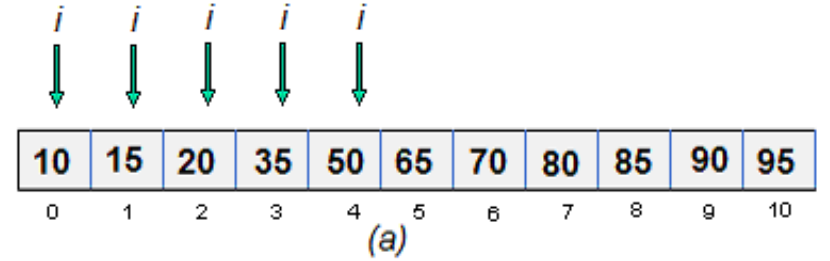
Pesquisa

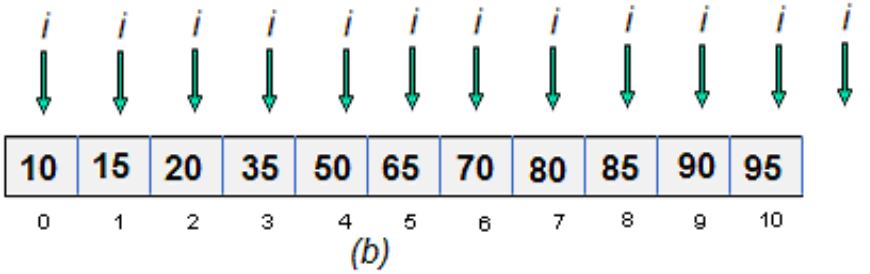
- **Overview**
  - Varredura: Pesquisa sequencial
    - Busca pelo valor 20



Pesquisa

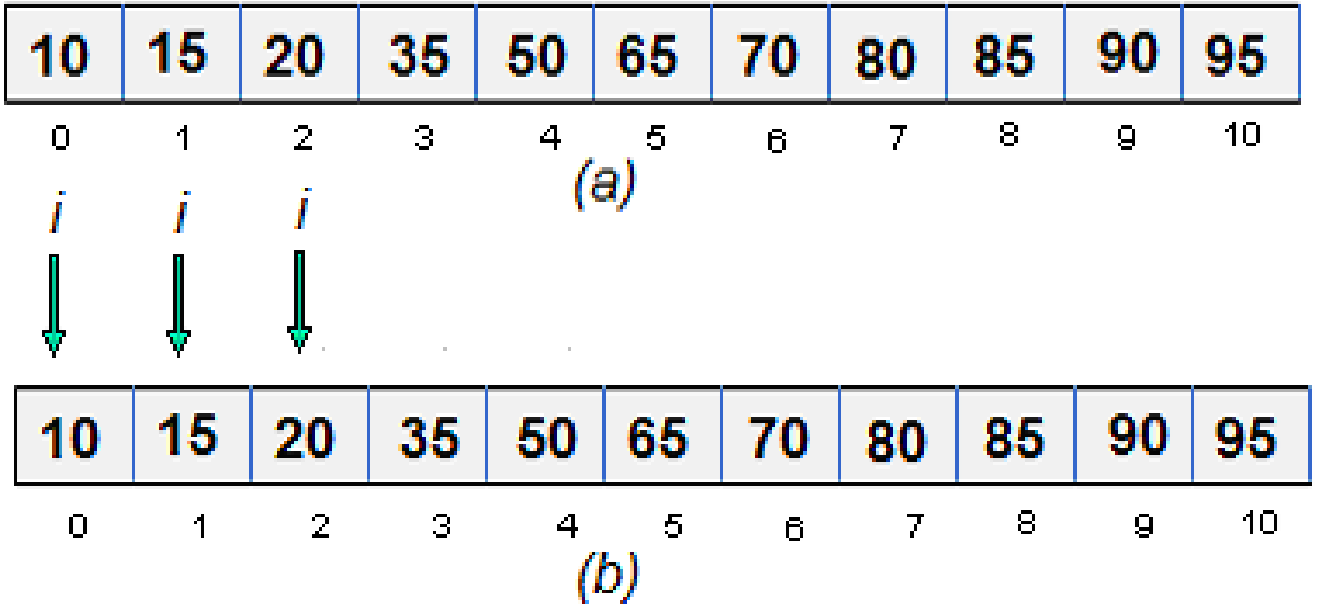
- **Overview**
  - Varredura: Pesquisa sequencial
    - Para a pesquisa da chave 40, deve-se varrer o arquivo e parar no endereço 4, pois, se a chave de pesquisa é menor que a chave corrente do vetor, e como o vetor está ordenado
    - Para a pesquisa da chave 100, deve-se varrer o arquivo e parar no final do vetor (endereço 12) .





Pesquisa

- **Overview**
  - Varredura: Pesquisa sequencial
    - **Missão**: implemente o código necessário para realizar a busca dos elementos 20, 40 e 100, no vetor abaixo:

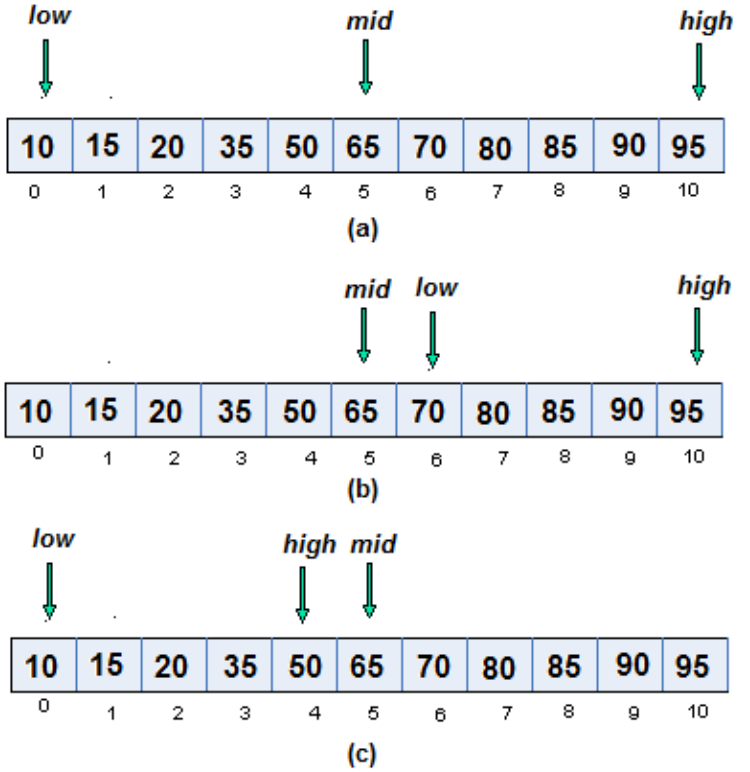


Pesquisa

- **Overview**
  - Pesquisa Binária
    - A idéia básica deste método está em visitar o registro do meio verificando se a chave de pesquisa é maior, menor ou igual à chave do arquivo.
    - Caso a chave seja igual, pare e retorne o endereço da célula.

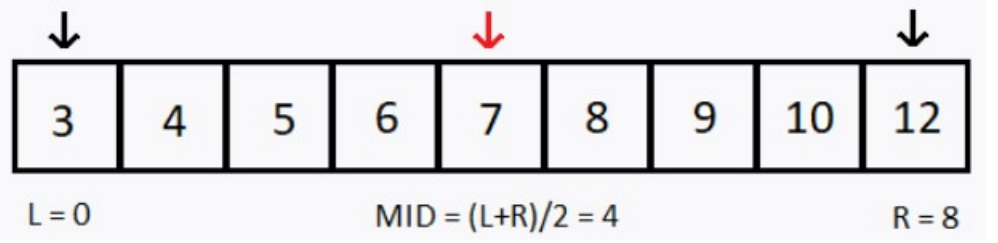
Pesquisa

- **Overview**
  - Pesquisa Binária
    - Supondo a chave de pesquisa igual a 65, neste caso a função retornará o endereço 5.
    - Caso a chave de pesquisa for maior que a chave do vetor, por exemplo 85, conclui-se que a chave deve estar no subvetor à direita da célula do meio.
      - Neste caso o ponteiro low receberá o valor mid+1, e a rotina é ativada novamente.
    - Caso a chave de pesquisa for menor que a chave do vetor, conclui-se que a chave deve estar no subvetor à esquerda da célula do meio. Neste caso o ponteiro high receberá o valor mid-1 e a rotina é ativada novamente.



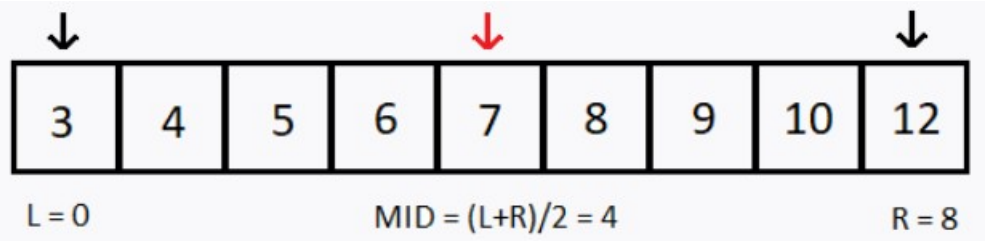
Pesquisa

- **Overview**
  - Pesquisa Binária
    - Procedimento de pesquisa binária.
    - 1. Início = 0, fim = tamanho do vetor
    - 2. Se inicio > fim parar o procedimento com insucesso.
    - 3. Visitar a célula do meio onde meio = quociente inteiro da divisão de início + fim por 2.
    - 4. Se a chave de pesquisa for igual à chave da célula do meio parar o procedimento com sucesso e retornar o endereço da célula do meio.
    - 5. Se a chave de pesquisa for menor que a chave do meio, ignorar todas as células posteriores á célula do meio, inclusive esta (basta fazer fim = meio - 1), e voltar ao passo 2.
    - 6. Se a chave de pesquisa for maior que a chave do meio, ignorar todas as células anteriores à célula do meio, inclusive esta (basta fazer início = meio +1), e voltar ao passo 2



Pesquisa

- **Overview**
  - Pesquisa Binária
    - **Missão**: implemente o código necessário para realizar a busca dos elementos 20, 40 e 100, no vetor abaixo:



Resultado das Missões

- Pesquisa Sequencial

```
global NOT_FOUND
NOT_FOUND=-1

# iterative scan search
def scan(alist,key):
  N = len(alist)
  for i in range(0, N):
      if key == alist[i]:
        return i
      elif key < alist[i]:
        return NOT_FOUND
  return NOT_FOUND
```

Resultado das Missões

- Pesquisa Binária

```
def iterativeBinarySearch(alist,key):
  N = len(alist)
  low = 0
  high = N-1
  mid=(low+high) //2
  while(low<=high):
    if key == alist[mid]:
      return mid
    elif key < alist[mid]:
      high=mid-1
    else:
      low=mid+1
    mid=(low+high) //2
  return NOT_FOUND
```

Atividades

- Implemente as pesquisas abaixo:
  - A) Pesquisa Sequencial
  - B) Pesquisa Binária

- Implemente um vetor com 30 números ordenados.
- Implemente um mecanismo que possa calcular o tempo de execução da pesquisa em sequencial e binária, comparando as duas, em termos de tempo gasto. Qual teve melhor desempenho de tempo?
- Faça a mesma análise de tempo com uma lista de 30 itens desordenada, sendo ordenada pelo bubbleSort e pelo quickSort. Qual o tempo melhor entre os 2, para ordenação + pesquisa binária?