

Engenharia de Software

Árvores e Gráfos

Aula 9: Algoritmos de busca sequencial x binária

Métodos de Busca: Algoritmos de Busca Sequencial X Busca Binária

- Importância em estudar busca
 - Busca é uma tarefa muito comum?
- Vários métodos e estruturas de dados podem ser empregados para se fazer busca
 - Quais estruturas de dados?
- Certos métodos de organização/ordenação de dados podem tornar o processo de busca mais eficiente

- O problema da busca (ou pesquisa)

“Dado um conjunto de elementos, onde cada um é identificado por uma chave, o objetivo da busca é localizar, nesse conjunto, o elemento que corresponde a uma chave específica”

- Tabela: termo genérico, pode ser qualquer estrutura de dados usada para armazenamento interno e organização dos dados
- Uma tabela é um conjunto de elementos, chamados registros

- Existe uma chave associada a cada registro, usada para diferenciar os registros entre si:
 - Chave interna: chave está contida dentro do registro, em uma localização específica
 - Chave externa: essas chaves estão contidas em uma tabela de chaves separada que inclui ponteiros para os registros
 - Chave primária: para todo arquivo existe pelo menos um conjunto exclusivo de chaves
 - Dois registros não podem ter o mesmo valor de chave
 - Chave secundária: são as chaves não primárias
 - Chaves que não precisam ter seus valores exclusivos

- Algoritmo de busca
- Formalmente, é o algoritmo que aceita um argumento **a** e tenta encontrar o registro cuja chave seja **a**

- Operações na tabela
 - Inserção: adicionar um novo elemento à tabela
 - Algoritmo de busca e inserção: se não encontra o registro, insere um novo
 - Remoção: retirar um elemento da tabela
 - Recuperação: procurar um elemento na tabela e, se achá-lo, torná-lo disponível

- A **tabela** pode ser:
 - Um vetor de registros
 - Uma lista encadeada
 - Uma árvore
 - Etc.
- A tabela pode ficar:
 - Totalmente na memória (busca interna)
 - Totalmente no armazenamento auxiliar (busca externa)
 - Dividida entre ambos

- Algumas técnicas de busca em memória interna são
 - Busca Seqüencial
 - Busca Binária
 - Busca por Interpolação
 - Busca em Árvores
 - Hashing
- O objetivo é encontrar um dado registro com o menor custo
 - Cada técnica possui vantagens e desvantagens

- A **busca seqüencial** é a forma mais simples de busca
 - É aplicável a uma tabela organizada como um vetor ou como uma lista encadeada

- Busca mais simples que há
 - Percorre-se registro por registro em busca

Sequential search

steps: 0



1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

www.penjee.com

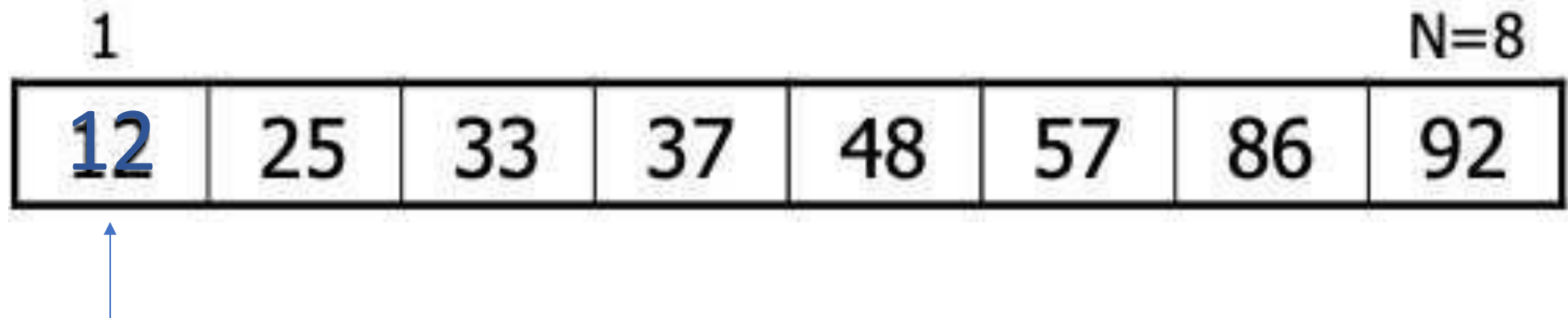
1							N=8
12	25	33	37	48	57	86	92

- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92

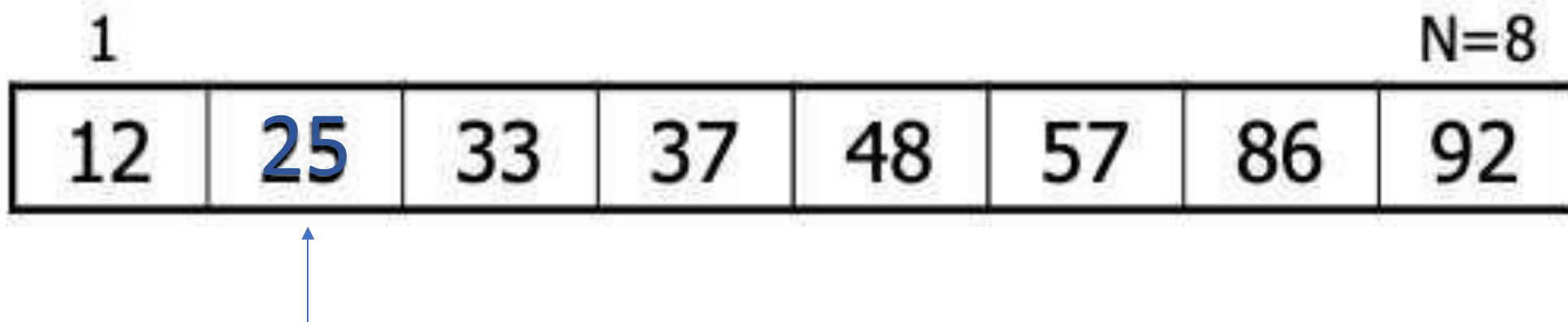
- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92



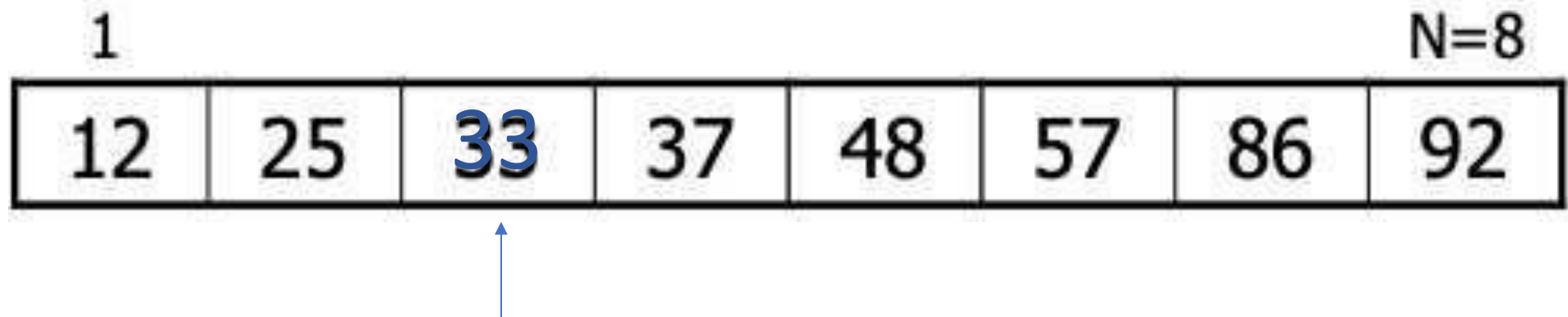
- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92



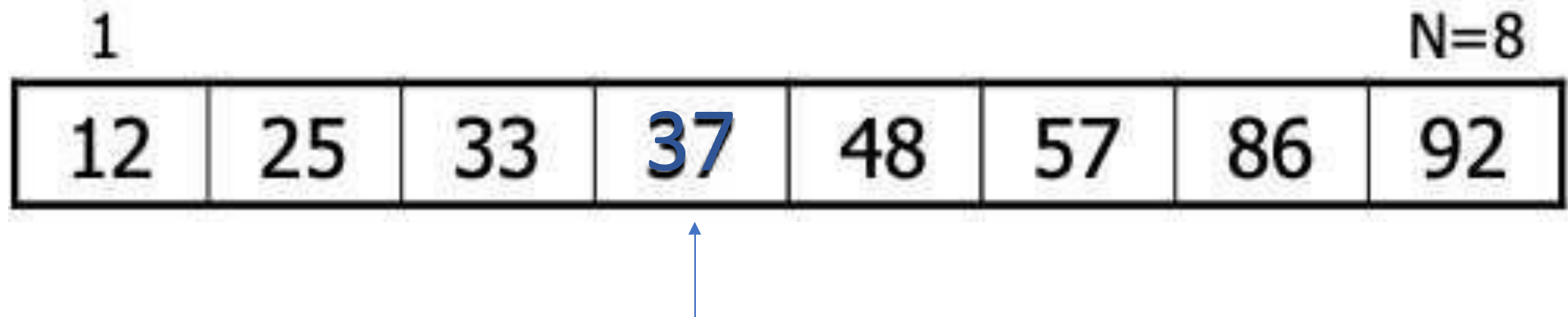
- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92



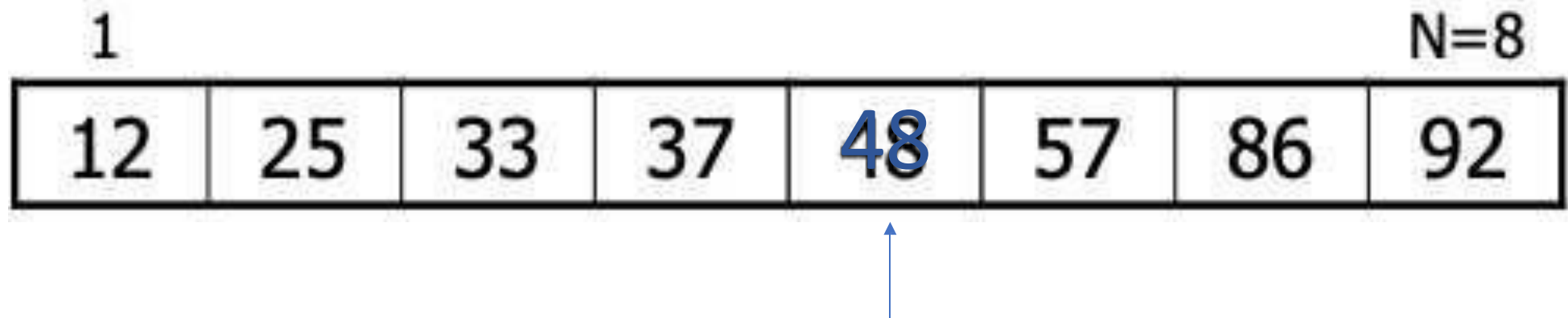
- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92



- Busca mais simples que há
 - Percorre-se registro por registro em busca da chave
- **Procure por 48**

1							N=8
12	25	33	37	48	57	86	92



- **Implementação**

- Algoritmo de busca seqüencial em um vetor A, com N posições (0 até N-1), sendo x a chave procurada

BUSCA-SEQUENCIAL[A, chave, N]

1. **para** $i = 1$ **até** N
2. **se** $A[i] = \text{chave}$ **então**
3. **retorna** $A[i]$
4. $i = i + 1$
5. **retorna** “chave não encontrada”

BUSCA BINÁRIA

- Se os dados estiverem ordenados em um arranjo, pode-se tirar vantagens dessa ordenação
 - Busca binária
 - $A[i] \leq A[i+1]$, se ordem crescente
 - $A[i] \geq A[i+1]$, se ordem decrescente

- O elemento buscado é comparado ao elemento do meio do arranjo
 - Se igual, busca bem-sucedida
 - Se menor, busca-se na metade inferior do arranjo
 - Se maior, busca-se na metade superior do arranjo

- Encontre o 25

Binary search

steps: 0

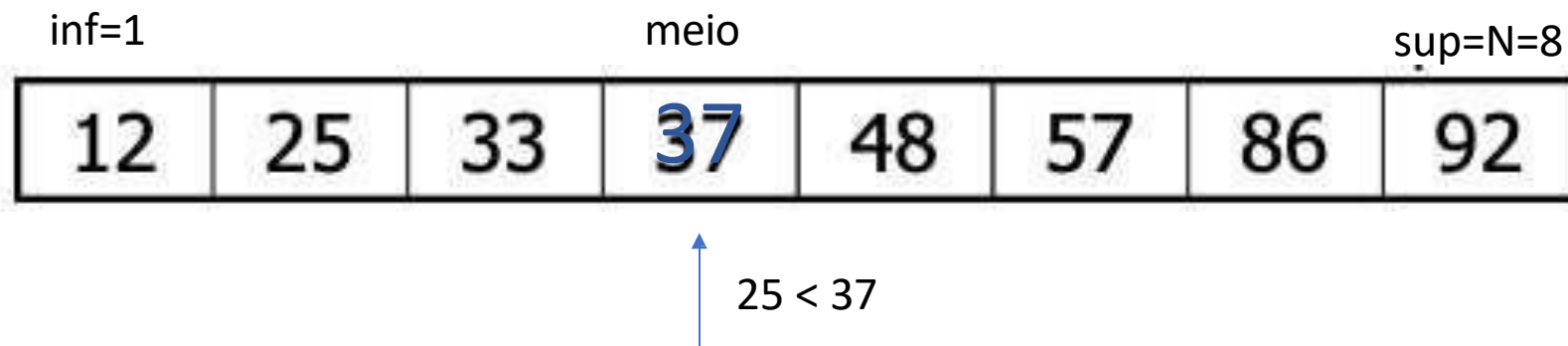


inf=1

meio

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----

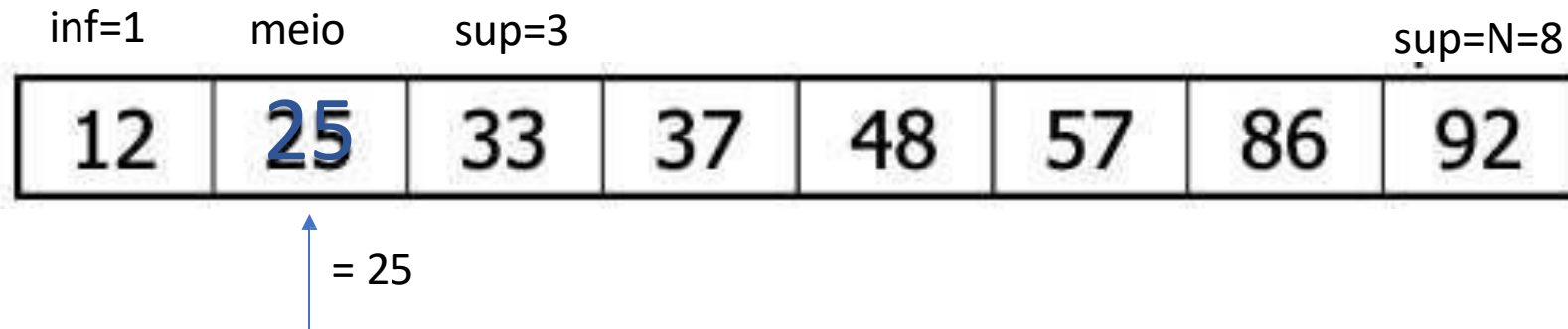
- Encontre o 25



- Encontre o 25

inf=1			sup=3				N=8
12	25	33	37	48	57	86	92

- Encontre o 25



- Em cada passo, o tamanho do arranjo em que se busca é dividido por 2

Exercício:

Escreva um programa que leia um vetor ordenado e utilize busca sequencial e busca binária para determinar qual valor está mais próximo da média do vetor.

Mostrar quantas comparações foram necessárias para encontrar o elemento com busca sequencial e busca binária

Vetor: 1 2 5 7 10 12 13 14 16 20

Pós Aula

- Escrever um programa que faça
 - 1) Busca sequencial
 - 2) Busca binária
- por um elemento em um arranjo ordenado e mostrar:
 - Quantos comparações foram necessárias para encontrar o elemento utilizando busca sequencial e a busca binária?
 - Erro se não existir o elemento no vetor
- Vetor: 1 2 3 4 5 5 7 8 9 10 12 14 17 22 35 39 41 44 47 48 49 51 55 54 57 87 89 99 100

