ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite

O problema da busca

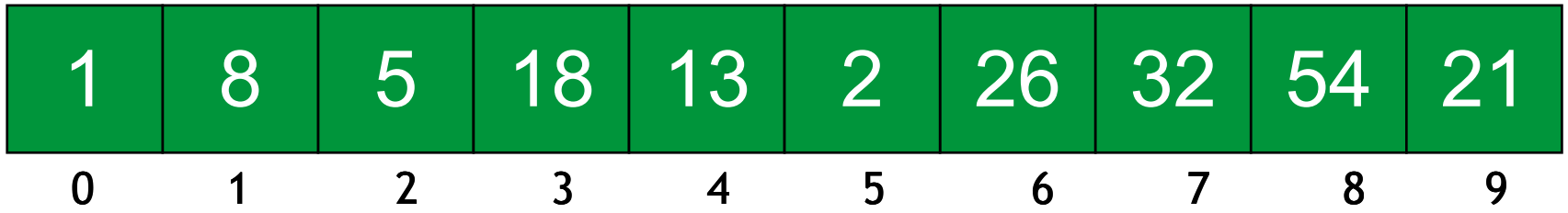
INSTITUTO FEDERAL
Pernambuco

- Definição: O problema de busca consiste em determinar se um elemento está presente em uma coleção de dados. Para isso, é fornecida uma chave de busca, que é comparada com as chaves dos elementos da coleção. Se houver correspondência, o elemento é localizado; caso contrário, conclui-se que ele não existe na coleção.

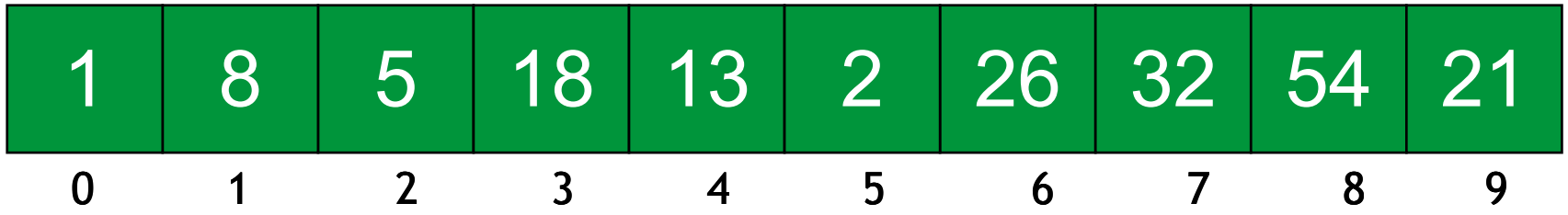
O problema da busca

INSTITUTO FEDERAL
Pernambuco

- key = 13



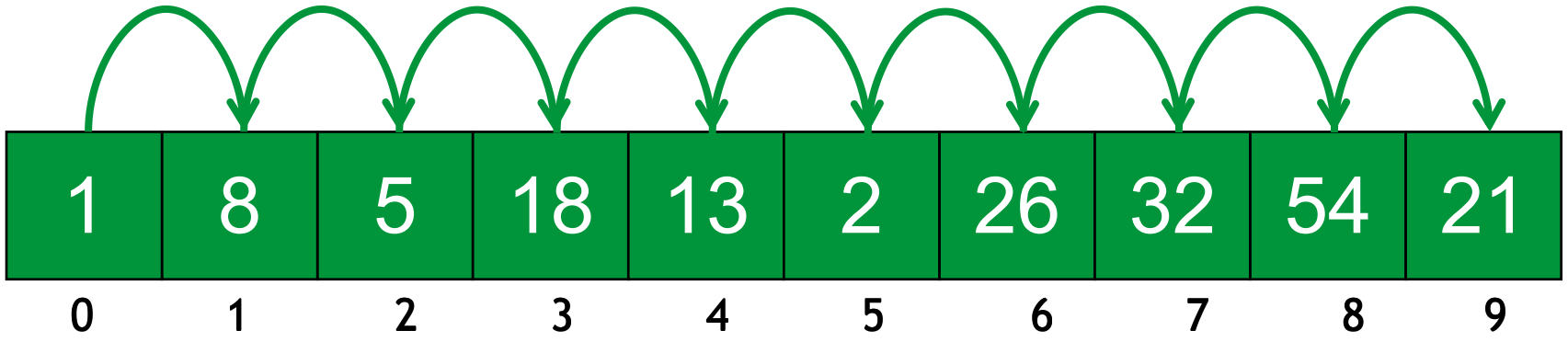
- key = 50



Busca Linear

INSTITUTO FEDERAL
Pernambuco

- Busca Linear (ou Busca Sequencial): É um algoritmo de busca que percorre os elementos da coleção sequencialmente, comparando cada elemento com a chave de busca até encontrar o valor desejado ou esgotar todos os elementos da coleção.



Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.

Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.
2. **O elemento atual é comparado com a chave de busca.**

Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.
2. O elemento atual é comparado com a chave de busca.
3. **Se o elemento corresponder à chave procurada, a busca é encerrada e a posição (índice) do elemento é retornada.**

Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.
2. O elemento atual é comparado com a chave de busca.
3. Se o elemento corresponder à chave procurada, a busca é encerrada e a posição (índice) do elemento é retornada.
4. **Caso contrário, o algoritmo avança para o próximo elemento da lista.**

Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.
2. O elemento atual é comparado com a chave de busca.
3. Se o elemento corresponder à chave procurada, a busca é encerrada e a posição (índice) do elemento é retornada.
4. Caso contrário, o algoritmo avança para o próximo elemento da lista.
5. **Os passos 2 a 4 são repetidos até que o elemento seja encontrado ou que todos os elementos da lista tenham sido examinados.**

Passos da Busca Linear

INSTITUTO FEDERAL
Pernambuco

1. O algoritmo inicia a busca a partir do primeiro elemento da lista.
2. O elemento atual é comparado com a chave de busca.
3. Se o elemento corresponder à chave procurada, a busca é encerrada e a posição (índice) do elemento é retornada.
4. Caso contrário, o algoritmo avança para o próximo elemento da lista.
5. Os passos 2 a 4 são repetidos até que o elemento seja encontrado ou que todos os elementos da lista tenham sido examinados.
6. **Se o fim da lista for alcançado sem que haja correspondência, a busca retorna um resultado indicando que o elemento não está presente na coleção (geralmente um valor especial como -1 ou null).**

Pseudocódigo da Busca Linear

INSTITUTO FEDERAL
Pernambuco

```
BuscaLinear(vetor, tamanho, chave)
    para i ← 0 até tamanho - 1 faça
        se vetor[i] == chave então
            retorne i
        fim-se
    fim-para

    retorne -1
```

Busca Linear em C

INSTITUTO FEDERAL
Pernambuco

```c
int buscaLinear(int v[], int n, int chave) {
    for (int i = 0; i < n; i++) {
        if (v[i] == chave)
            return i;
    }

    return -1;
}
```

Busca Linear

INSTITUTO FEDERAL
Pernambuco

- Como a busca linear pode necessitar percorrer todos os elementos da coleção, seu tempo de execução é proporcional ao tamanho da entrada, resultando em uma complexidade de O(n).
- Vantagens:
  - Simplicidade de implementação.
  - Versatilidade: pode ser usada em qualquer tipo de coleção, seja ela ordenada ou não.
- Desvantagens:
  - Ineficiente para coleções grandes em comparação com outros algoritmos de busca mais avançados, como a busca binária.
  - Complexidade linear.

Busca Linear

- Como melhorar a Busca Linear?

Busca Linear

INSTITUTO FEDERAL
Pernambuco

- Como melhorar a Busca Linear?
  - Se o vetor estiver ordenado, é possível melhorar a busca linear interrompendo a execução assim que for encontrado um elemento maior que a chave procurada.
  - Como os elementos estão em ordem crescente, a chave não poderá aparecer nas posições seguintes.

Busca Linear com vetor ordenado em C

INSTITUTO FEDERAL
Pernambuco

```c
int buscaLinearOrdenada(int vetor[], int tamanho, int chave) {
    for (int i = 0; i < tamanho; i++) {

        if (vetor[i] == chave)
            return i;

        if (vetor[i] > chave)
            return -1;
    }

    return -1;
}
```

Busca Binária

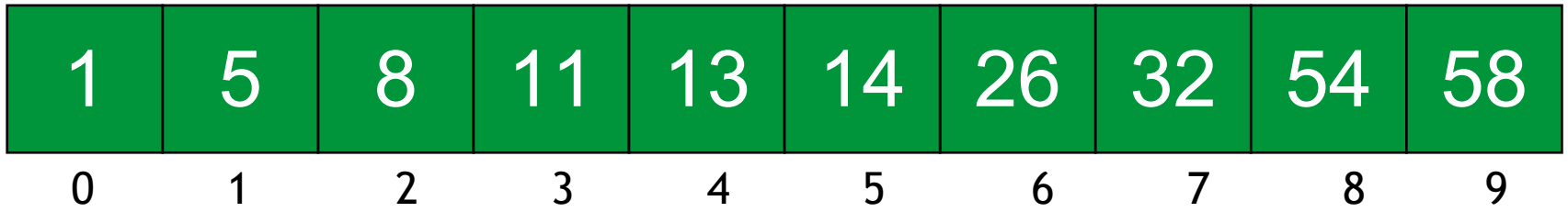
INSTITUTO FEDERAL
Pernambuco

- A busca binária é um algoritmo eficiente para localizar um elemento em uma coleção ordenada.
- Em vez de examinar os elementos um a um, como ocorre na busca linear, ela reduz o espaço de busca pela metade a cada comparação.
- O algoritmo compara a chave procurada com o elemento localizado no meio da coleção. Se houver correspondência, a busca é encerrada. Caso contrário, apenas a metade que pode conter a chave continua sendo considerada, enquanto a outra metade é descartada.

O problema da busca

INSTITUTO FEDERAL
Pernambuco

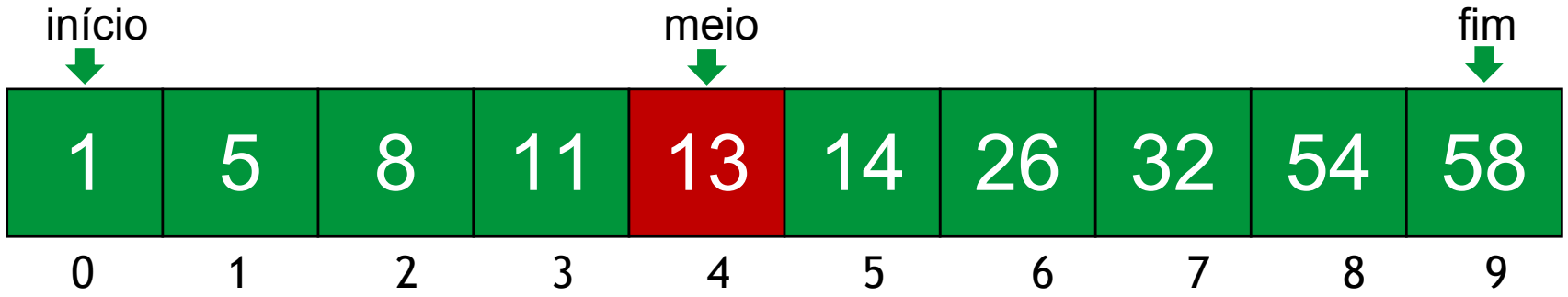
- key = 54



O problema da busca

INSTITUTO FEDERAL
Pernambuco

- key = 54

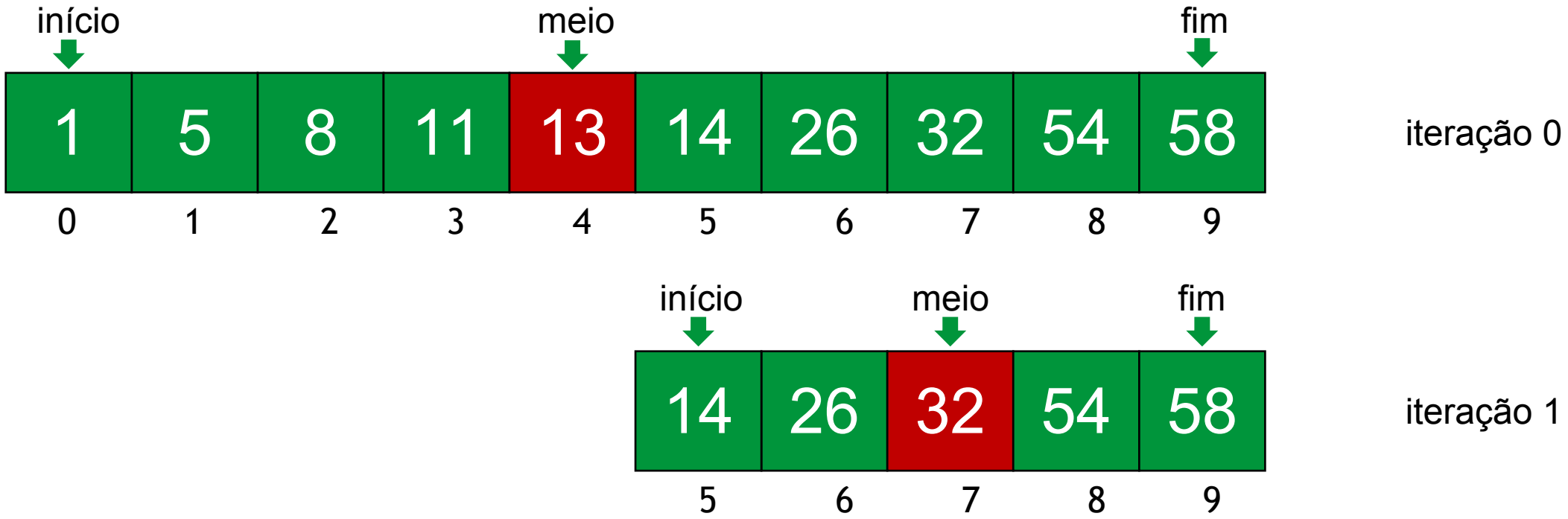


iteração 0

O problema da busca

INSTITUTO FEDERAL
Pernambuco

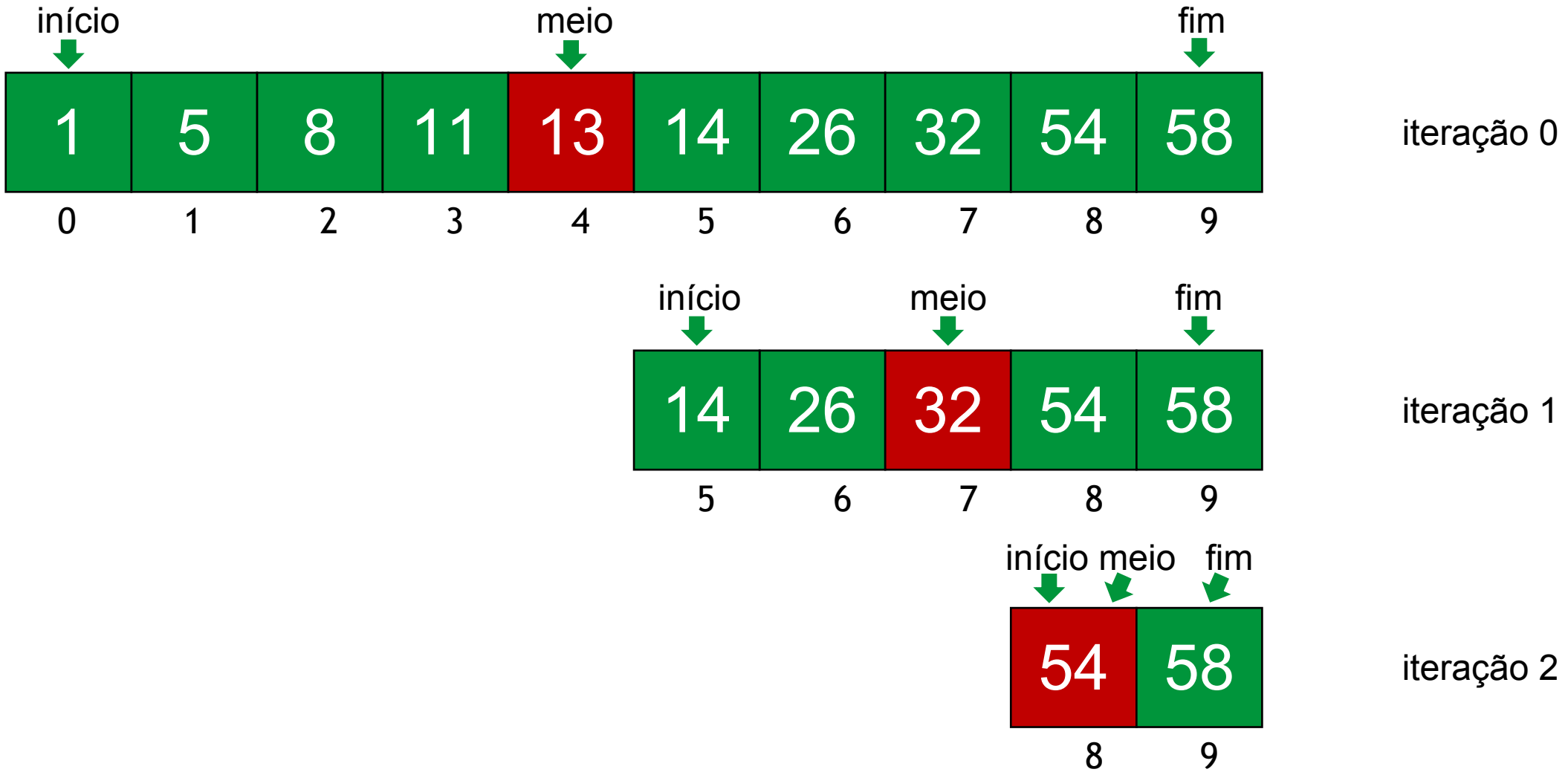
- key = 54



O problema da busca

INSTITUTO FEDERAL
Pernambuco

- key = 54



Busca Binária

INSTITUTO FEDERAL
Pernambuco

- Devido a estratégia de divisão sucessiva do espaço de busca, a busca binária possui complexidade de tempo O(log n), onde n representa a quantidade de elementos da coleção, tornando-a significativamente mais eficiente que a busca linear em coleções grandes.

Observação: A busca binária só pode ser aplicada em coleções cujos elementos estejam previamente ordenados.

Passos da Busca Binária

INSTITUTO FEDERAL
Pernambuco

1. Inicialize dois índices: início, apontando para o primeiro elemento da lista, e fim, apontando para o último elemento.

Passos da Busca Binária

INSTITUTO FEDERAL
Pernambuco

1. Inicialize dois índices: início, apontando para o primeiro elemento da lista, e fim, apontando para o último elemento.
2. **Calcule o índice do elemento central da lista (ou da porção da lista que ainda está sendo considerada na busca).**

Passos da Busca Binária

INSTITUTO FEDERAL
Pernambuco

1. Inicialize dois índices: início, apontando para o primeiro elemento da lista, e fim, apontando para o último elemento.
2. Calcule o índice do elemento central da lista (ou da porção da lista que ainda está sendo considerada na busca).
3. **Compare o elemento central com a chave procurada:**
   - **Se forem iguais, a busca é encerrada e o índice do elemento é retornado.**
   - **Se a chave for menor que o elemento central, descarte a metade direita da lista ajustando fim = meio - 1.**
   - **Se a chave for maior que o elemento central, descarte a metade esquerda da lista ajustando início = meio + 1.**

Passos da Busca Binária

INSTITUTO FEDERAL
Pernambuco

3. Compare o elemento central com a chave procurada:
   - Se forem iguais, a busca é encerrada e o índice do elemento é retornado.
   - Se a chave for menor que o elemento central, descarte a metade direita da lista ajustando fim = meio - 1.
   - Se a chave for maior que o elemento central, descarte a metade esquerda da lista ajustando início = meio + 1.
4. **Recalcule o índice do elemento central considerando apenas a nova faixa definida pelos índices início e fim.**

Passos da Busca Binária

INSTITUTO FEDERAL
Pernambuco

3. Compare o elemento central com a chave procurada:
   - Se forem iguais, a busca é encerrada e o índice do elemento é retornado.
   - Se a chave for menor que o elemento central, descarte a metade direita da lista ajustando fim = meio - 1.
   - Se a chave for maior que o elemento central, descarte a metade esquerda da lista ajustando início = meio + 1.
4. Recalcule o índice do elemento central considerando apenas a nova faixa definida pelos índices início e fim.
5. **Repita os passos 3 e 4 enquanto início ≤ fim. Se, em algum momento, início > fim, a busca retorna um resultado indicando que o elemento não está presente na coleção (geralmente um valor especial como -1 ou null).**

Pseudocódigo da Busca Binária

INSTITUTO FEDERAL
Pernambuco

```
BuscaBinaria(vetor, inicio, fim, chave)

    se fim < inicio então
        retorne -1

    meio ← [(inicio + fim)/2]

    se vetor[meio] = chave então
        retorne meio

    senão se chave < vetor[meio] então
        retorne BuscaBinaria(vetor, inicio, meio - 1, chave)

    senão
        retorne BuscaBinaria(vetor, meio + 1, fim, chave)
```

Busca Binária em C

INSTITUTO FEDERAL
Pernambuco

```c
int buscaBinaria(int vetor[], int inicio, int fim, int chave) {
    if (fim < inicio)
        return -1;

    int meio = (inicio + fim) / 2;

    if (vetor[meio] == chave)
        return meio;
    else if (chave < vetor[meio])
        return buscaBinaria(vetor, inicio, meio - 1, chave);
    else
        return buscaBinaria(vetor, meio + 1, fim, chave);
}
```

Busca Binária em C

INSTITUTO FEDERAL
Pernambuco

```c
int buscaBinaria(int vetor[], int inicio, int fim, int chave) {
    if (fim < inicio)
        return -1;

    int meio = (inicio + fim) / 2;

    if (vetor[meio] == chave)
        return meio;
    else if (chave < vetor[meio])
        return bu
    else
        return bu
}
```

meio ← [(inicio + fim)/2] pode causar overflow se inicio e fim forem valores muito grandes.

Correção para evitar overflow:
int meio = inicio + (fim - inicio) / 2;

Busca Binária em C (solução iterativa)

```c
int buscaBinaria(int vetor[], int tamanho, int chave) {
    int inicio = 0;
    int fim = tamanho - 1;

    while (inicio <= fim) {
        int meio = (inicio + fim) / 2;

        if (vetor[meio] == chave)
            return meio;

        if (chave < vetor[meio])
            fim = meio - 1;
        else
            inicio = meio + 1;
    }

    return -1;
}
```

INSTITUTO FEDERAL
Pernambuco

Busca Binária

- Vantagens:
  - Alta eficiência: A cada comparação, metade dos elementos é descartada, reduzindo rapidamente o espaço de busca e exigindo poucas comparações, mesmo em coleções grandes.
  - Complexidade logarítmica: Possui complexidade de tempo O(log n), sendo muito mais eficiente que a busca linear (O(n)) em coleções grandes.
- Desvantagens:
  - Pouco adequada para listas encadeadas: Requer acesso rápido ao elemento central, algo não disponível nesse tipo de estrutura.
  - Requer dados ordenados.
  - Implementação mais complexa.

Busca Linear vs Busca Binária

INSTITUTO FEDERAL
Pernambuco

Característica	Busca Binária	Busca Linear
Estratégia	Divisão e conquista	Força bruta
Complexidade	O(log n)	O(n)
Requer ordenação	Sim	Não
Estruturas mais adequadas	Vetores ou coleções com acesso por índice	Qualquer tipo de coleção
Implementação	Mais complexa	Simples
Desempenho em grandes coleções	Excelente	Limitado

Busca Linear vs Busca Binária

INSTITUTO FEDERAL
Pernambuco

- Quando utilizar?
  - Busca Binária: indicada para coleções grandes e ordenadas, quando a eficiência é um requisito importante.
  - Busca Linear: indicada para coleções pequenas, não ordenadas ou quando a simplicidade da implementação é mais importante que o desempenho.

Exercícios

INSTITUTO FEDERAL
Pernambuco

1. Os algoritmos de busca apresentados nesta aula foram implementados para vetores. Adapte-os para funcionarem em uma lista encadeada simples, considerando a seguinte estrutura:

```
typedef struct No {
    int valor;
    struct No *prox;
} No;
```

a) Adapte a busca linear.

b) Adapte a busca linear para uma lista ordenada.

c) É possível adaptar a busca binária para uma lista encadeada? Justifique sua resposta.

Dúvidas

INSTITUTO FEDERAL
Pernambuco

ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite