Análise de Algoritmos

Correção de Algoritmos

Invariante do Laço

- A maioria dos algoritmos consomem até 90% do tempo de execução dentro de laços. Logo, os laços computam a maior parte da solução do problema.
- Saber exatamente o que cada um dos laços no algoritmo faz e que o faz corretamente é extremamente importante para provar que o algoritmo é correto.
- Um laço executa sempre o mesmo conjunto de instruções e por isso faz sempre a mesma coisa, então o que acontece na execução do laço não muda, não varia, é invariante.

Invariante do Laço

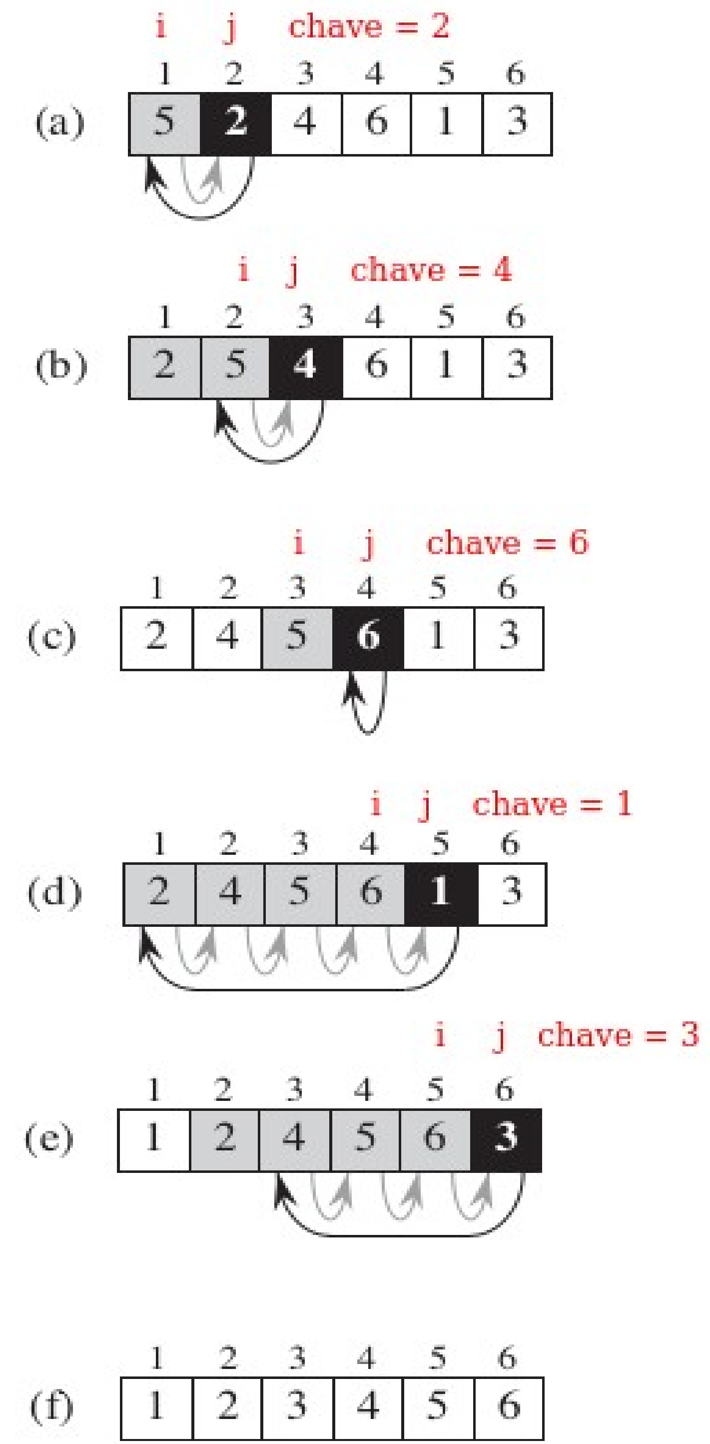
- Chamamos de **Invariante do Laço** uma descrição do que o laço em questão executa, apresentando o que sempre acontece **em função das variáveis, instruções e dados internos**.
- Não basta ser uma descrição corrida da funcionalidade executada pelo laço porque para provar que o Invariante acontece para todas as entradas é necessário, através de uma indução, apresentar que o invariante é verdadeiro antes da 1ª iteração, durante todas as demais iterações e após o encerramento do laço.

Prova de Correção

- Para demostrar que o laço está correto é preciso mostrar que o invariante é sempre verdadeiro nas seguintes situações:
  - **Inicialização**: (antes da primeira iteração)
  - **Manutenção**: (durante as demais iterações)
  - **Finalização**: (após o encerramento das iterações)

Exemplo

```
Algoritmo INSERTION-SORT(A)
  para j = 2 até A.tamanho
    chave = A[ j ]
    i = j - 1
    enquanto i > 0 e A[ i ] > chave faça
      A[ i+1] = A[ i ]
      i = i - 1
    fim_enquanto
    A[ i + 1] = chave
  fim_para
fim_algoritmo
```



Exemplo

- **Qual o Invariante do laço 1-9** ?

  - O que podemos afirmar de concreto sobre os dados antes da 1ª iteração ?

  – O que podemos afirmar de concreto sobre os dados antes de cada uma das demais iterações ? A resposta à primeira pergunta se mantém ?

  – O que podemos afirmar de concreto sobre os dados após o encerramento do laço ? As respostas para as perguntas anteriores influenciam nessa situação ?

Exemplo

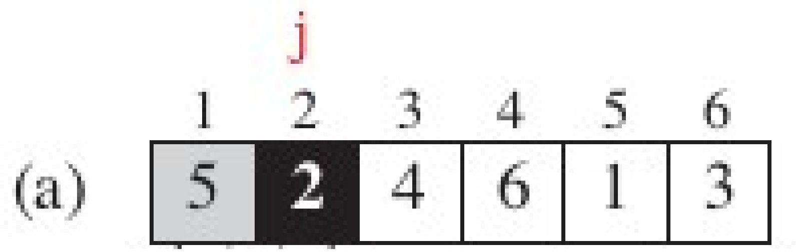
Invariante do Laço 1-9:

No inicio de cada iteração do laço, o subvetor A[1 .. j-1] consiste dos elementos originalmente em A[1 .. j-1], mas ordenados.

Verdade ?

Como isso garante que nosso algoritmo atinge seu objetivo sempre ?

Prova de Correção do laço 1-9



Inicialização:

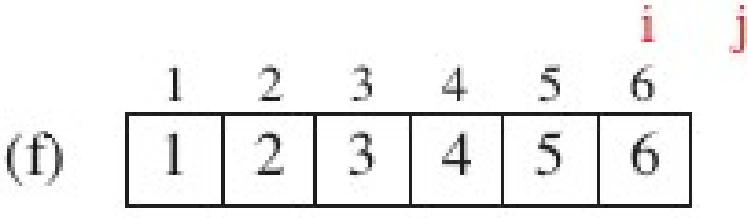
- antes da primeira iteração do laço, j=**2**.
- assim, A[1 .. j-1] = A[1..**2**-1],
- portanto, A[1..1] = A[1],
- que é de fato o elemento original em A[1], e obviamente está ordenado,
- logo, o invariante do laço é válido antes da primeira iteração.

Prova de Correção do laço 1-9

Manutenção:

- **Informalmente**, o corpo do laço **para** funciona movendo A[j - 1], A[j-2], A[j- 3], e assim por diante por uma posição para a direita até encontrar a posição adequada para chave(A[j]) (linhas 4–7), onde ele insere o valor de chave(A[j]) (linha 8). O subvetor A[1 .. j] consiste então nos elementos originalmente em A[1 .. j], mas em ordem classificada. Incrementar j para a próxima iteração do laço **para** preserva o invariante.
- **Formalmente**, é preciso definir e provar a corretude do laço **enquanto** (linhas 4-7). - Exercício.

Prova de Correção do laço 1-9



Terminação:

- O laço **para** encerra quando j > A.tamanho (n), ou seja, j=**n+1**,
- logo, o subvetor A[1..j-1] = A[1..**n+1**-1] = A[1..n],
- então, no final temos o vetor inteiro com os elementos originais ordenados.

Exercício

- Determine o **Invariante do laço** enquanto (4-7) do algoritmo Insertion-Sort e prove sua correção.