Universidade de Brasília
Departamento de Ciência da Computação
Projeto e Análise de Algoritmos

Atividade 06

José Antônio Alcântara da Silva de Andrade Mat: 232013031

Professor:
Flávio Leonardo Calvacanti de Moura
Turma 02

Brasília, DF

20 de maio de 2025

Algoritmo Partition

Listagem 1: Algoritmo Partition

```
1  x <- A[r];
2  i <- p - 1;
3  FOR j = p TO r - 1 DO
4      IF A[j] <= x THEN
5          i <- i + 1;
6          EXCHANGE A[i] WITH A[j];
7      END
8  END
9  EXCHANGE A[i+1] WITH A[r];
10 RETURN i + 1;
```

Exercício

O exercício consiste em provar a seguinte invariante de laço para o algoritmo da Listagem 1:

> Antes de cada iteração do laço **FOR** (linhas 3-8), para todo k, temos:
>
> 1. Se $p \leq k \leq i$, então $A[k] \leq x$;
> 2. Se $i + 1 \leq k \leq j - 1$, então $A[k] \geq x$;
> 3. Se $k = r$, então $A[k] = x$.

Primeiro Item

Podemos reescrever o problema da seguinte forma: note o subvetor $A[p..i]$. Todos os elementos dele serão, necessariamente, menores que o valor x. Provaremos, então, que isso se permanece em toda iteração.

Antes da iteração inicial ($j = p$) teremos que $A[p..i] = A[p..p - 1]$ é um subvetor nulo; satisfazendo as condições.

Ocorre um de dois casos em cada iteração: ou $A[j] \leq x$ ou $A[j] > x$. No caso $A[j] > x$, o valor de i segue inalterado e, por hipótese de indução, $A[p..i]$ permanece satisfazendo as condições.

No caso $A[j] \leq x$, ocorrerá uma troca dos elementos $A[j]$ e $A[i+1]$. Como o elemento $A[j]$ é garantidamente menor ou igual x, teremos que o subvetor $A[p..i + 1]$ satisfaz as condições. Ainda mais, como o valor de i é incrementado nesse caso, na manuntenção dessa iteração, o vetor $A[p..i]$ seguirá válido.

Portanto, independente do caso, o subvetor $A[p..i]$ consistirá apenas de elementos menores ou iguais a x no início de toda iteração.

Segundo Item

Reescrevendo o item, devemos provar que o subvetor $A[i+1..j-1]$ consiste apenas de elementos estritamente maiores que x.

Antes da iteração inicial ($j = p$) teremos que $A[i+1..j-1] = A[p..p-1]$ é um subvetor nulo; satisfazendo as condições.

Ocorre um de dois casos em cada iteração: ou $A[j] \leq x$ ou $A[j] > x$. No caso $A[j] > x$, não há alterações nas variáveis ou vetor original, portanto, o subvetor $A[i+1..j]$ terá apenas elementos maiores que x. Isso, por sua vez, garante manuntenção da invariância, quando j for incrementado e o subvetor $A[i+1..j-1]$ continuar com apenas elementos maiores que x.

No caso $A[j] \leq x$, abrem-se duas novas possibilidades: $i+1 = j$ e $i+1 < j$ (o valor de i sempre será menor ou igual a j). No caso em que $i+1 = j$, o vetor $A[i+1..j-1] = A[j..j-1]$ é um vetor nulo, satisfazendo as condições independente das troca de $A[i+1]$ com $A[j]$.

No caso $i+1 < j$, teremos que o elemento $A[i+1]$ é, necessariamente, maior que x antes da troca, uma vez que o subvetor $A[i+1..j-1]$ satisfaz as condições por hipótese de indução. Contudo, o elemento $A[j] \leq x$, então, ao trocá-lo com $A[i+1]$, o subvetor válido será deslocado em uma unidade para $A[i+2..j]$. Como o valor de i é incrementado nesse caso, e o valor de j é incrementado no final da iteração, as condições se mantém satisfeitas na manuntenção, onde o subvetor $A[i+1..j-1]$ terá apenas elementos maiores que x.

Terceiro Item

Note que o intervalo de valores de j é $p \leq j \leq r-1$. Ou seja, j nunca será igual a r, o que então implica que, durante as iterações do loop, o elemento $A[r]$ permanece em posição. Sendo assim, $A[r] = x$ durante toda a iteração.