Cálculo da complexidade do Algoritmo de Floyd Modificado

Autor: João Vicente Souto

Disciplina: INE5408 - Estrutura de Dados - UFSC

Pseudocódigo Floyd Modificado:

```
Início
    Para i = 1 até n faça
        Para j = 1 até n faça
            A[i,j] <- D[i,j];
            R[i,j] <- 0;

    Para i = 1 até n faça
        A[i,i] <- 0;

    Para k = 1 até n faça
        Para i = 1 até n faça
            Para j = 1 até n faça
                Se A[i,k] + A[k,j] < A[i,j] então faça
                    A[i,j] <- A[i,k] + A[k,j];
                    R[i,j] <- k;
Fim
```

Análise:

Considerações:

- **Cálculo será dividido nos 3 laços independentes do algoritmo e somados no final.**
- Cada operação de atribuição, aritmética e comparação custam uma unidade de tempo.
- Laços executam n vezes suas operações internas, n em função do laço.
- *Cálculo é efetuado de dentro para fora em laços e funções.*
- Extra: Funções normais (sem considerar as recursivas) custam as unidades de suas operações internas.

Primeiro laço principal

```
Para i = 1 até n faça
    Para j = 1 até n faça
        A[i,j] <- D[i,j];
        R[i,j] <- 0;
```

Dentro do laço interno:

- 2 atribuições;
- *Executado n vezes*

Laço interno:

- 1 atribuição de j;
- n+1 comparações de j e n;
- n incrementos de j;
- *Executado n vezes.*

Laço principal:

- 1 atribuição de i;
- n+1 comparações de i e n;
- n incrementos de i.

1º Cálculo: `n*(n*2 + n+(n+1)+1) + n+(n+1)+1` = `4n^2 + 4n + 2`

Segundo laço principal

```
Para i = 1 até n faça
    A[i,i] <- 0;
```

Dentro do laço:

- 1 atribuições;
- *Executado n vezes.*

Laço:

- 1 atribuição de i;
- n+1 comparações de i e n;
- n incrementos de i.

2º Cálculo: `n*1 + n+(n+1)+1` = `3n + 2`

Terceiro laço principal

```
Para k = 1 até n faça
    Para i = 1 até n faça
        Para j = 1 até n faça
            Se A[i,k] + A[k,j] < A[i,j] então faça
                A[i,j] <- A[i,k] + A[k,j];
                R[i,j] <- k;
```

Dentro do laço interno:

- 2 atribuições;
- 2 somas;
- 1 comparação;
- *Executado n vezes;*
- *Considerando o pior caso em que (A[i,k] + A[k,j] < A[i,j]) sempre seja verdadeiro.*

Laço interno:

- 1 atribuição de j;
- n+1 comparações de j e n;
- n incrementos de j;
- *Executado n vezes.*

Laço intermediário:

- 1 atribuição de i;
- n+1 comparações de i e n;
- n incrementos de i;
- *Executado n vezes.*

Laço principal:

- 1 atribuição de k;
- n+1 comparações de k e n;
- n incrementos de k;

3º Cálculo: `n*(n*(n*5 + n+(n+1)+1) + n+(n+1)+1) + n+(n+1)+1` = `7n^3 + 4n^2 + 4n + 2`

Conclusão

Somando-se todas as 3 partes do algoritmo que foram calculadas em separado, segue que: Cálculo final: `4n^2 + 4n + 2` + `3n + 2` + `7n^3 + 4n^2 + 4n + 2` = `7n^3 + 8n^2 + 11n + 6`

Considerando o comportamento assintótico do algoritmo, ignorando os elementos de menor peso e a constante que multiplica o elemento de maior peso, conclui-se que o algoritmo de floyd é `O(n^3)`.