

Nome: Rafael Bicalho Roth

Matrícula: 2019100082

Turma: TW

Documentação - Trabalho Prático

Como primeiro passo, separei a leitura do arquivo em 3 fases:

1. Armazenar o tamanho: É a primeira coisa a ser feita, pois será necessário ter o tamanho para alocar espaço no nosso array e matriz.
2. Armazenar os nós: Para isso, foi criada uma estrutura "Element" com as propriedades "index" e "value", para identificar o valor do nó e sua posição na matriz adjacente.
3. Criar e preencher uma matriz adjacente: Para representar as ligações, foi criada uma matriz adjacente, onde as linhas eram a origem e as colunas o destino. Essa matriz foi preenchida usando uma função para buscar a posição do elemento de acordo com seu valor .

Após termos a matriz adjacente completa, foram criadas funções, uma para cada tipo de relação binária, que percorriam a matriz e faziam as comparações necessárias entre suas posições para verificar as propriedades da relação binária do conjunto.

É importante destacar:

1. Na função criada para verificar se a relação binária é reflexiva e na função criada para verificar se a relação binária é irreflexiva, foi necessário criar somente um laço para acessar os valores da matriz adjacente. Isso só é necessário verificar as ligações onde a origem e destino são iguais. Consequentemente, o pior caso de complexidade e tempo de execução se dá por $O(n)$, onde n são o número de nós.
2. Nas funções para verificar se a relação binária é simétrica, anti-simétrica, assimétrica e transitiva, uma para cada, foi criar dois laços para acessar os valores da matriz. Porém pelo fato de se tratar de uma matriz adjacente, é possível acessar todos os valores da matriz apenas por um de seus triângulos, reduzindo na metade o números de iterações. Apesar de o

número de interações ser reduzido potencialmente, o pior caso de complexidade e tempo de execução se dá por $O(n^2)$, onde n são o número de nós.

3. Já na função para verificar se a relação binária é transitiva, foi necessário criar 3 laços, um dentro do outro, para acessar os valores necessários. Uma curiosidade, é que em nenhum dos casos foi necessário imprimir as ligações, nesta função, por exemplo, a cada vez que uma ligação que tornaria a relação transitiva era encontrada, o valor das colunas era zerado e o valor da linha incrementado. Contudo, essa função representa o pior caso de complexidade e tempo de execução de todo o programa, e se dá por $O(n^3)$, onde n são o número de nós.

As demais verificações foram feitas comparando o resultado das funções mencionadas.