

SCC0503 - Algoritmos e Estruturas de Dados II -

Trabalho II - Grafos

Daniel De Marco Fucci

NUSP: 11218639

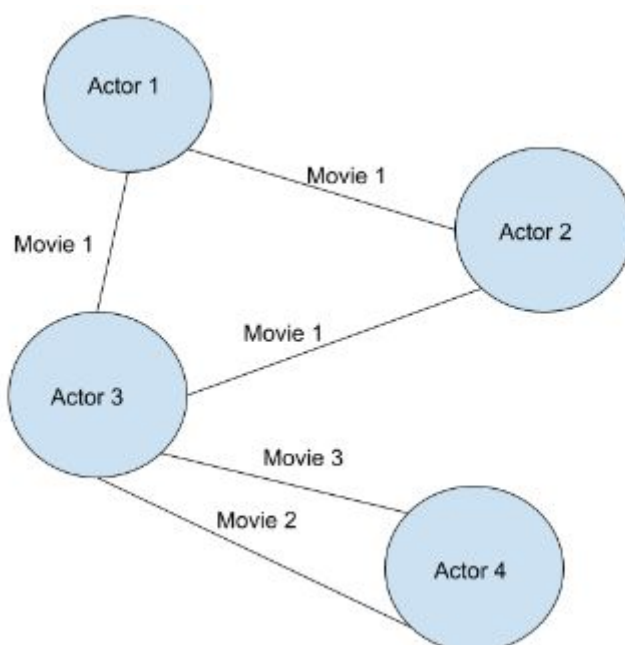
Pedro Fernando Christofolletti dos Santos

NUSP: 11218560

Relatório:

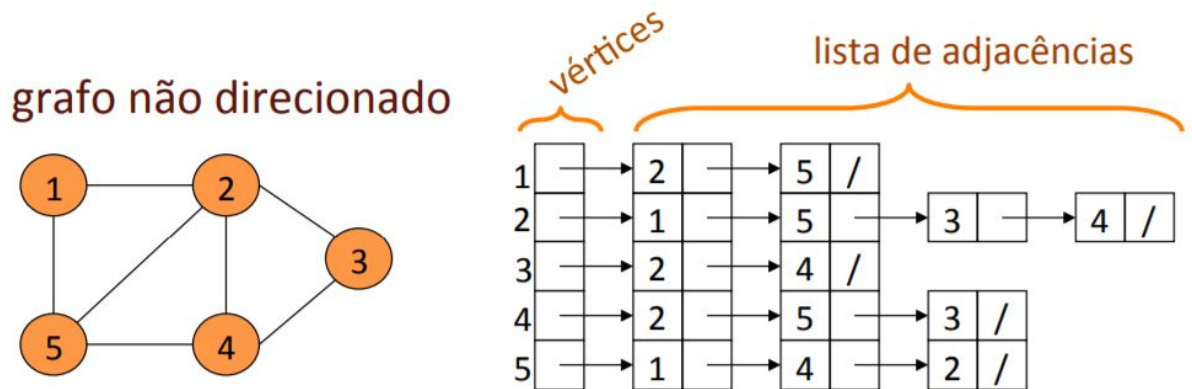
1. Abordagem: utilizamos um grafo com as seguintes características:
 - a. Listas de adjacência: a representação permite flexibilidade de espaço, não é necessário saber previamente o número de vértices e se aplica melhor ao grafo esparso do problema.
 - b. Ponderado: as arestas possuem peso, que nesse caso é o nome do filme que forma a conexão entre os atores.
 - c. Não-direcional: Se o ator X atuou com ator Y, logo o ator Y atuou com o ator X
 - d. Arestas múltiplas: Um ator pode ter trabalhado com outro ator em mais de um filme, gerando duas arestas distintas entre eles, como é o caso dos atores 3 e 4 do exemplo abaixo, que trabalharam juntos nos filmes “Movie 2” e “Movie 3”, gerando duas arestas entre eles.
 - e. Não-conexo, pois os vértices não são todos conectados entre si diretamente.

Representação do grafo



2. Leitura de dados e inserção de vértices e arestas:

A aplicação lê o arquivo linha por linha(filme por filme) e para cada ator do filme, verifica se o ator já existe no vetor de atores, caso não exista, ele é adicionado ordenadamente como um novo índice no vetor, em que cada índice representa um ator e uma lista encadeada, representando os atores com quem ele trabalhou.



Sendo assim, para cada filme, cada ator é conectado com os demais com o peso da arestas sendo o filme com o qual trabalharam juntos.

3. Cálculo do número de Kevin Bacon e registro do caminho percorrido:

Para realizar esse cálculo e registrar o caminho percorrido de um determinado ator até Kevin Bacon, o algoritmo realiza busca em largura, com os seguintes passos:

- Busca binária pelo ator informado no vetor de atores(para cada ator temos uma lista encadeada mostrando os atores com quem ele trabalhou)
- Adiciona o ator informado(ator inicial) em uma fila
- Enquanto a fila não estiver vazia: retira um ator da fila(na primeira iteração esse é o ator inicial), verifica em sua lista de adjacência se Kevin Bacon está presente(se trabalharam juntos em algum filme), sendo que para cada elemento da lista de adjacência verificado, caso não seja Kevin Bacon é verificado em uma estrutura auxiliar se esse ator já foi verificado, caso não tenha sido, ele é adicionado na fila e marcado como já visitado na estrutura auxiliar(state). Então o processo se repete retirando da fila o próximo ator
- Para cada ator retirado da fila, ele também é adicionado em um vetor de rastreio, que guarda qual ator está sendo verificado(teve sua lista de adjacência verificada a procura de Kevin Bacon) e qual o ator antecessor. Todos os atores com exceção do inicial(informado pelo

usuário) foram adicionados na fila pois estavam presentes na lista de adjacência de algum outro ator(atuou com outro ator), o ator antecessor é justamente o dono da lista de adjacência em que o ator que está sendo verificado estava presente e por isso foi adicionado.

- e. Ao encontrar Kevin Bacon percorrendo os atores e listas de adjacências, a lista de rastreio é convertida para uma lista encadeada, possibilitando inverter o sentido (Em vez de ser sentido “Kevin Bacon -> Ator inicial” transformamos para “Ator inicial -> Kevin Bacon”) e então imprimimos o caminho percorrido do ator inicial até KB.

4. Estrutura do código: A modularização do trabalho foi feita usando 9 arquivos, sendo eles:

“**main.c**”: Arquivo principal para compilar o código e controle de ações para navegação do usuário. Basicamente esse arquivo controla as ações de um Menu usando cases de um Switch para cada opção, sendo elas:

1- Read file and create graph

Lê o arquivo de input disponibilizado e cria um grafo com suas informações.

2- Search Actor/Actress KB's number

Pede ao usuário que entre com o nome de um(a) ator/atriz e mostra tanto o seu número de Kevin Bacon quanto o menor caminho que o levou até ele mostrando os filmes e atores/atrizes intermediários(as).

3- Average KB's number and standard deviation

Calcula a média e o desvio padrão dos números de Kevin Bacon (com exceção dos infinitos) de todos os(as) atores/atrizes do grafo.

4- Free graph and exit

Libera toda a memória alocada e finaliza a execução.

“**grafos.c**”: Arquivo que contém todas as funções e integrações utilizadas para realizar as operações disponíveis aos usuários. Além disso, o arquivo difere as funções internas das que são disponibilizadas ao usuário e estão contidas no arquivo de header. As funções internas começam com o prefixo “__”. Ademais, esse arquivo utiliza de funções contidas em outros dois arquivos “stack.c” e “queue.c” a fim de garantir o funcionamento e clareza do código, evitando um aglomerado de funções.

“grafos.h”: Arquivo de cabeçalho contendo as structs e defines usados pelo arquivo “grafos.c” e também o cabeçalho das funções que serão disponibilizadas para o arquivo “main.c”.

“queue.c”: Arquivo que contém as funções básicas para o funcionamento de uma fila, na qual os elementos são adicionados sempre no último lugar da fila e retirados a partir do primeiro da fila. Utilizamos dessas funções no arquivo “grafos.c” para efetuar a busca em largura.

“queue.h”: Arquivo de cabeçalho contendo as structs e defines usados pelo arquivo “queue.c” e também o cabeçalho das funções que serão disponibilizadas para o arquivo “grafos.c”.

“stack.c”: Arquivo que contém as funções básicas para o funcionamento de uma pilha, no qual os elementos são adicionados sempre no topo e retirados do topo. Utilizamos dessas funções no arquivo “grafos.c” para o armazenamento do caminho na busca em largura.

“stack.h”: Arquivo de cabeçalho contendo as structs e defines usados pelo arquivo “stack.c” e também o cabeçalho das funções que serão disponibilizadas para o arquivo “grafos.c”.

“Makefile.c”: Arquivo que consta as instruções de como gerar o arquivo binário a ser compilado.

“input-top-grossing.txt”: Arquivo de input disponibilizado para criar o grafo. Em cada linha do arquivo contém o nome do filme seguido dos(as) atores/atrizes que trabalharam no mesmo, separados por “/”.