

Análise e Síntese de Algoritmos

1º Projeto - 24 de março de 2017

77906 António Sarmento 84711 Diogo Redin

Introdução

Neste projeto foi-nos pedido que criássemos um programa que ordenasse cronologicamente um dado número de fotografias, sendo dadas as suficientes relações cronológicas entre as mesmas. O objetivo do projeto consistiu em escolher uma estrutura de dados adequada e um algoritmo que nos permitisse ordenar as fotografias, sempre que possível.

Para representar as relações entre as fotografias, decidimos utilizar um grafo dirigido, em que cada vértice representa uma fotografia e cada arco consiste numa relação cronológica entre as mesmas (i.e., se a fotografia A acontece antes que a fotografia B, no grafo representamos a relação como A -> B). Com esta implementação, encontrar uma ordenação cronológica das fotografias, traduz-se em encontrar uma ordenação topológica do grafo.

Os principais desafios deste problema consistiram em saber quando as relações dadas entre as fotografias eram insuficientes ou incoerentes.

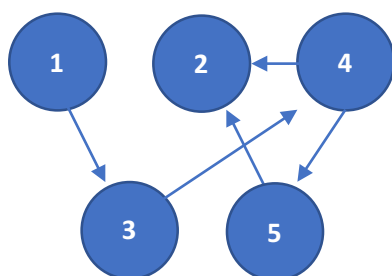
Análise Teórica

Como estrutura de dados para representar o grafo dirigido, escolhemos um vetor de arcos, recorrendo a três vetores:

1. Tamanho: No. de vértices (V); Índice: Número do vértice; Conteúdo: Índice do arco;
2. Tamanho: No. de arcos (E); Índice: Número do arco; Conteúdo: Vértice Final;
3. Tamanho: No. de arcos (E); Índice: Número do arco; Conteúdo: Arco Irmão;

Recorremos ainda a dois vetores adicionais de tamanho igual ao número de vértices para guardar o grau de cada vértice e o resultado da ordenação topológica.

Exemplo da representação de um Grafo Dirigido:



#	Vértice
1	1
2	0
3	2
4	3
5	5

#	Arco	Irmão
1	3	-
2	4	-
3	2	4
4	5	-
5	2	-

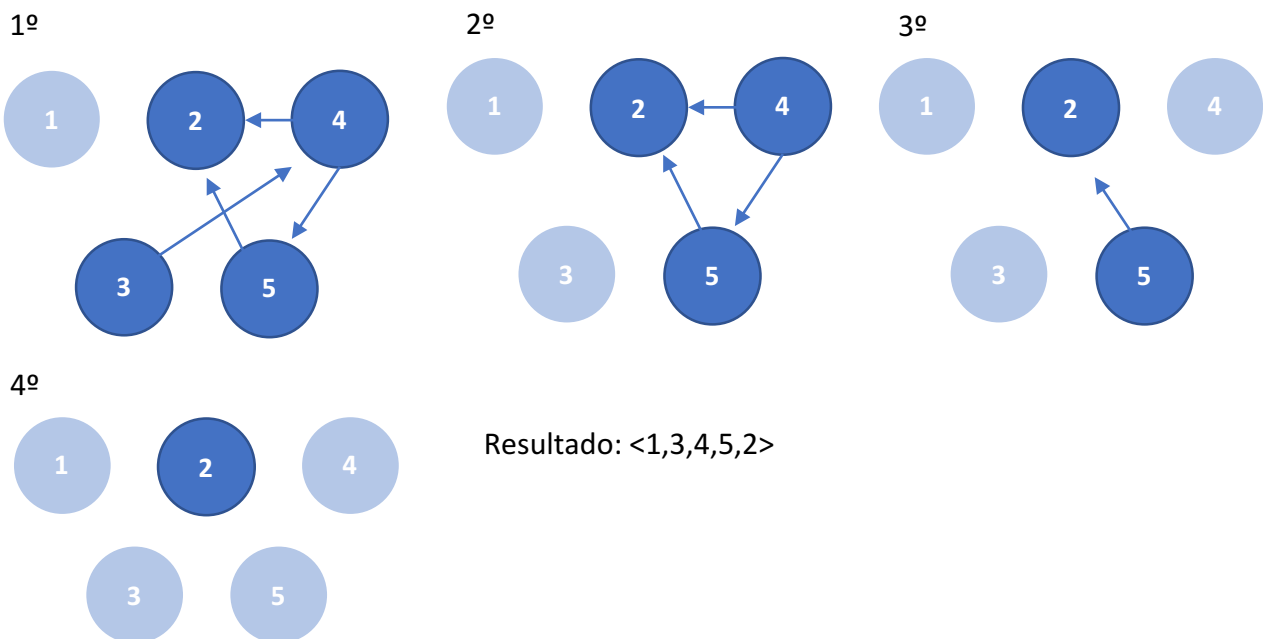
#	Grau
1	0
2	2
3	1
4	1
5	1

A nossa estrutura de dados tem, portanto, a seguinte eficiência para cada operação:

- Espaço: $O(E)$
- Inicialização: $O(1)$
- Inserir Arco: $O(1)$
- Encontrar Arco: $O(E)$

Para fazer a ordenação topológica do grafo, implementámos o algoritmo de Khan. Este algoritmo consiste em:

1. Colocar todos os vértices sem arcos de chegada (grau 0) numa pilha;
2. Criar um contador de vértices visitados que começa a 0;
3. Remover um dos vértices da pilha e:
 - a. Guardar o vértice no vetor de resultados;
 - b. Incrementar em 1 o contador dos vértices visitados;
 - c. Decrementar em 1 o grau de todos os vértices filhos;
 - d. Se o grau de algum dos vértices filhos passar a ser 0 adicioná-lo à pilha;
4. Repetir o passo 3 até não haver vértices com grau 0 na pilha;



Output: “Insuficiente”

Sabemos que não há relações suficientes para haver uma única ordenação topológica quando na execução do passo 3.c. um vértice tem mais do que um vértice filho com grau 0.

Output: “Incoerente”

Sabemos que há relações inconsistentes quando o número de vértices no resultado é diferente do número de vértices dados, o que aconteceria se algum vértice aparecesse mais do que uma vez na ordenação, o que não é possível.

O algoritmo tem eficiência $O(V + E)$ uma vez que percorremos todos os vértices uma vez para descobrir quais têm grau 0 (1.), e posteriormente percorremos todos os arcos existentes (3.d.).

Resultados