

O TAD GRAPH

Dept. de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

José Diogo Cerqueira, Bernardo Marujo
(76758) c.jose.diogo@ua.pt, (107322) bernardomarujo@ua.pt

[Link para o Repositório](#)

5 de janeiro de 2024



Conteúdo

1	Análise da complexidade dos três algoritmos de ordenação topológica.	1
1.1	Primeiro Algoritmo	1
1.1.1	Descrição	1
1.1.2	Análise da Complexidade	1
1.2	Segundo Algoritmo	2
1.2.1	Descrição	2
1.3	Terceiro Algoritmo	2
1.3.1	Descrição	2
1.4	Fatores de Influência dos Algoritmos	3
1.5	Dados Experimentais	3
1.5.1	Grafo Orientado Acíclico (grafo do ficheiro DAG_4.txt)	3
1.5.2	Grafo Orientado com Peso (grafo do ficheiro SWmediumEWD.txt)	4

Análise da complexidade dos três algoritmos de ordenação topológica.

1.1 Primeiro Algoritmo

1.1.1 Descrição

Este algoritmo consiste em utilizar um loop para encontrar vértices sem arestas de entrada no grafo. A cada iteração, é adicionado o vértice encontrado à sequência topológica e são removidas as suas arestas de saída no grafo cópia. A função `GraphTopoSortComputeV1` opera sobre um grafo direcionado representado pela estrutura de dados `Graph`. No entanto, é crucial observar que este algoritmo não é adequado para grafos que contenham ciclos.

1.1.2 Análise da Complexidade

Para analisar a complexidade do algoritmo, adotamos as seguintes métricas

- **remEdgeOps (Operações de Remoção de Arestas)** - Conta o número total de operações para remover as arestas de saída dos vértices na cópia do grafo.
- **copyGraphOps (Operações de Cópia do Grafo)** - Conta o número total de operações para criar uma cópia do grafo original.
- **outerLoopIter (Iterações do Loop Externo)** - Conta o número total de iterações do loop externo, responsável por encontrar vértices sem arestas de entrada.
- **innerLoopIter (Iterações do Loop Interno)** - Conta o número total de iterações do loop interno, usado para percorrer todos os vértices em busca daqueles sem arestas de entrada.

Complexidade da Execução

O loop externo é executado no máximo o número de vértices do grafo.

O loop interno é executado no máximo o número de vértices do grafo em cada iteração do loop externo.

As operações dentro do loop interno (verificar grau de entrada, marcar vértices, remover arestas, etc.) são operações constantes.

Portanto, a complexidade de execução pode ser expressa por $O(\text{numVertices} \times \text{numVertices}) = O(\text{numVertices}^2)$.

1.2 Segundo Algoritmo

1.2.1 Descrição

O algoritmo V2 implementa uma ordenação topológica usando uma abordagem baseada numa busca em profundidade (DFS - *Depth-First Search*). Ele começa por encontrar um vértice não visitado, explorando o máximo possível ao longo de cada ramo antes de retroceder.

Para analisar a complexidade do algoritmo, adotamos as seguintes métricas

- **DFScomps (Operações na Função DFS)** - Conta o número total de operações em todas as chamadas à função DFS. Representa a complexidade em termos de profundidade do grafo explorado.
- **adjacentsOps (Operações em Vértices Adjacentes)** - Conta o número total de operações relacionadas a vértices adjacentes durante o DFS. Inclui a obtenção de adjacentes e verificações de visitação.
- **sortingOps (Operações de Ordenação)** - Conta o número total de operações de ordenação realizadas para construir a ordenação topológica com base nos tempos de finalização do DFS.

Complexidade da Execução

A função DFS é chamada uma vez para cada vértice não visitado do grafo. O custo de uma chamada DFS é $O(\text{numVertices} + \text{numArestas})$, onde numArestas é o número de arestas no grafo.

A construção da ordenação topológica com base nos tempos de término da DFS tem complexidade $O(\text{numVertices})$.

A etapa subsequente de construção da ordenação topológica contribui com um fator $\log(V)$ adicional, devido à ordenação.

A complexidade total do algoritmo V2 é dominada pela componente DFS, resultando numa ordem $O(V + E)$.

1.3 Terceiro Algoritmo

1.3.1 Descrição

O algoritmo V3 adota a abordagem de Breadth-First Search (BFS) para realizar a ordenação topológica.

A BFS utiliza uma fila para controlar a ordem de visita dos vértices. A fila segue o princípio de "primeiro a entrar, primeiro a sair" (FIFO), onde novos vértices são inseridos no final da fila (enqueue), e a remoção ocorre no início (dequeue). Esta estrutura de dados é fundamental para garantir que os vértices sejam visitados de acordo com a ordem em que foram descobertos.

No contexto do algoritmo V3, o enqueue é utilizado para adicionar novos vértices à fila durante a busca, garantindo que sejam processados na ordem correta. A dequeue é empregada para obter o próximo vértice a ser explorado.

Para analisar a complexidade do algoritmo, adotamos as seguintes métricas

- **BFSOps (Operações durante o BFS)** - Representa o número total de operações realizadas durante a BFS.
- **queueEnqOps (Operações de enqueue)** - Número de operações de adição de um elemento à fila durante a BFS.
- **queueDeqOps (Operações de dequeue)** - Número de operações de remoção do elemento mais antigo da fila durante a BFS.

Complexidade da Execução

O número de operações BFSOps representa a complexidade do algoritmo. Para um grafo com V vértices e E arestas, a complexidade média esperada é da ordem de $O(V + E)$. Esta complexidade é associada ao processo de explorar todos os vértices e arestas do grafo.

O algoritmo V3 apresenta uma abordagem eficiente para a ordenação topológica de grafos orientados, utilizando a BFS para explorar o grafo de maneira nivelada. As métricas fornecem *insights* sobre o desempenho do algoritmo em diferentes contextos e cenários.

Tempo: A complexidade de tempo está principalmente associada ao número total de operações BFSOps. Num grafo com V vértices e E arestas, a complexidade é aproximadamente $O(V + E)$, pois cada vértice e cada aresta são visitados uma vez.

Espaço: O uso da fila na BFS contribui para a complexidade espacial. A complexidade de espaço é $O(V)$ para armazenar a fila e as estruturas adicionais.

1.4 Fatores de Influência dos Algoritmos

- :
- Estrutura do Grafo: A densidade e a estrutura do grafo afetam o número de operações. Grafos mais densos ou com ciclos podem aumentar a complexidade.
- Tamanho do Grafo: O número de vértices e arestas influencia diretamente o desempenho. Grandes grafos exigem mais operações.
- Marcação de Vértices: A marcação de vértices visitados é essencial para evitar a repetição de visitas e a ocorrência de loops infinitos, contribuindo assim para uma melhor eficiência do algoritmo.

1.5 Dados Experimentais

1.5.1 Grafo Orientado Acíclico (grafo do ficheiro DAG_4.txt)

Métrica	Algoritmo		
	V1	V2	V3
time	0.000025	0.000005	0.000005
caltime	0.000015	0.000003	0.000003
remEdgeOps	25	0	0
copyGraphOps	40	0	0
outerLoopIter	15	0	0
innerLoopIter	120	0	0
DFScomps	0	25	0
adjacentsOps	25	25	25
sortingOps	0	15	0
BFSOps	0	0	25
queueEnqOps	0	0	15
queueDeqOps	0	0	15
<i>number of operations</i>	200	65	55

Tabela 1.1: Resultados dos Testes para Algoritmos de Ordenação Topológica

Resultado V1: 0 2 1 6 5 4 3 14 13 12 11 10 9 8 7
Resultado V2: 0 2 6 14 13 5 12 11 1 4 10 9 3 8 7
Resultado V3: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

1.5.2 Grafo Orientado com Peso (grafo do ficheiro SWmediumEWD.txt)

Métrica	Algoritmo		
	V1	V2	V3
time	0.001111	0.000086	0.000090
caltime	0.000680	0.000052	0.000055
remEdgeOps	0	0	0
copyGraphOps	2796	0	0
outerLoopIter	250	0	0
innerLoopIter	62500	0	0
DFScomps	0	2546	0
adjacentsOps	0	2546	2546
sortingOps	0	250	0
BFSOps	0	0	2546
queueEnqOps	0	0	250
queueDeqOps	0	0	250
<i>number of operations</i>	65546	5342	3046

Tabela 1.2: Resultados dos Testes para Algoritmos de Ordenação Topológica

Resultado V1: * The topological sorting could not be computed!! ***** (o grafo apresenta ciclos)

Resultado V2: 0 15 24 39 66 149 80 44 49 59 50 48 45 3 37 31 115 76 95 153 228 241 83 67 5 4 26 55 78 62 21 27 65 7 42 2 14 18 35 12 28 41 17 81 53 22 34 56 73 120 119 64 29 47 91 109 38 74 126 183 137 215 134 145 146 218 227 224 167 117 6 16 54 99 129 13 19 70 30 8 11 43 82 85 152 143 105 10 106 84 79 51 86 108 101 110 122 92 132 154 142 9 23 33 58 68 114 163 202 93 32 52 77 102 138 151 57 118 124 155 165 191 97 144 104 185 201 216 232 248 168 160 176 204 231 187 208 226 222 209 211 225 206 217 171 172 180 213 235 238 195 245 125 148 71 135 141 94 198 113 90 173 69 107 1 72 150 40 20 75 89 127 61 87 130 194 116 164 190 169 46 161 177 186 189 220 247 200 158 121 170 182 223 249 229 203 242 36 88 98 178 236 166 133 100 103 174 179 212 156 139 196 205 207 210 244 246 123 175 214 219 221 181 157 184 188 233 240 239 112 128 136 159 234 230 197 131 193 243 192 162 140 147 111 25 60 63 96 199 237

Resultado V3: 0 15 24 44 49 58 59 68 80 97 114 149 160 163 176 191 202 204 209 211 222 225 39 66 206 93 144 168 185 231 248 9 23 33 50 165 32 187 52 226 48 104 201 232 151 208 142 195 118 124 155 171 172 180 213 5 77 102 26 138 45 83 216 217 57 65 154 235 238 245 184 197 92 132 125 4 55 67 78 240 239 21 27 62 188 233 3 76 95 7 148 157 181 230 71 122 139 205 101 112 128 159 136 90 37 115 153 228 241 42 108 135 196 110 156 207 210 214 219 221 234 69 173 87 113 242 31 2 86 141 51 79 212 30 43 8 11 82 152 244 70 61 130 107 111 194 12 28 121 158 170 182 198 223 94 14 18 35 133 19 84 174 179 143 85 246 175 100 89 1 72 200 203 25 60 20 40 75 116 164 220 36 41 88 17 229 249 129 166 13 103 192 243 106 131 193 105 10 123 127 150 189 177 186 63 96 199 237 190 247 169 98 81 134 53 120 161 6 16 99 147 178 236 162 46 117 119 146 227 64 91 137 145 22 34 56 73 54 140 167 29 47 109 218 224 183 215 38 74 126