

Resumo

Foi implementado em java o algoritmo de caminhos disjuntos. Os testes foram realizados em 3 grafos de tamanhos diferentes e o tempo de execução do algoritmo foi medido utilizando a classe *System.currentTimeMillis()*;

Grafos

Foram utilizados 3 grafos, sendo que o Grafo 1 contém 50 vértices, o Grafo 2, 250 vértices e o Grafo 3, 500 vértices.

Resultados

A Tabela 1 mostra os resultados de eficácia e eficiência do algoritmo, sendo que a eficácia é medida pela taxa de acerto dos caminhos disjuntos encontrados no grafo e a eficiência é medida pelo tempo de execução do algoritmo em milissegundos. A eficiência foi medida utilizando a classe *System.currentTimeMillis()* do Java.

	Caminho	Nº de caminhos disjuntos esperados	Nº de caminhos disjuntos encontrados	Eficácia	Eficiência (tempo de execução em ms)
Grafo 1	0 -> 49	10	10	100%	18
	5 -> 27	10	10	100%	17
	45 -> 12	7	7	100%	20
	31 -> 48	6	6	100%	19
Grafo 2	0 -> 249	7	7	100%	79
	54 -> 138	8	8	100%	85
	112 -> 5	4	4	100%	81
	45 -> 230	11	11	100%	87
Grafo 3	0 -> 499	11	11	100%	163
	357 -> 12	14	14	100%	178
	25 -> 46	8	8	100%	156
	130 -> 375	15	15	100%	175

Tabela 1 - eficácia e eficiência do algoritmo

A Tabela 2 mostra os caminhos disjuntos encontrados pelo algoritmo em cada caminho específico.

Caminho	Resultado do algoritmo
0 -> 49	0 -> 19 -> 49 0 -> 1 -> 48 -> 49 0 -> 4 -> 3 -> 49 0 -> 10 -> 15 -> 49 0 -> 16 -> 35 -> 49 0 -> 29 -> 32 -> 49 0 -> 39 -> 20 -> 49 0 -> 47 -> 46 -> 49 0 -> 6 -> 5 -> 26 -> 49 0 -> 41 -> 32 -> 31 -> 49
5 -> 27	5 -> 9 -> 27 5 -> 14 -> 27 5 -> 26 -> 27 5 -> 28 -> 27 5 -> 4 -> 3 -> 27

	5 -> 6 -> 40 -> 27 5 -> 12 -> 46 -> 27 5 -> 39 -> 38 -> 27 5 -> 7 -> 8 -> 43 -> 27 5 -> 35 -> 36 -> 44 -> 27
45 -> 12	45 -> 46 -> 12 45 -> 19 -> 10 -> 12 45 -> 36 -> 32 -> 12 45 -> 38 -> 40 -> 12 45 -> 42 -> 8 -> 12 45 -> 24 -> 6 -> 5 -> 12 45 -> 44 -> 10 -> 11 -> 12
31 -> 48	31 -> 49 -> 48 31 -> 11 -> 10 -> 48 31 -> 20 -> 15 -> 48 31 -> 22 -> 3 -> 48 31 -> 39 -> 1 -> 48 31 -> 12 -> 11 -> 47 -> 48
0 -> 249	0 -> 51 -> 35 -> 249 0 -> 1 -> 45 -> 46 -> 249 0 -> 7 -> 190 -> 124 -> 249 0 -> 211 -> 105 -> 248 -> 249 0 -> 112 -> 44 -> 230 -> 164 -> 249 0 -> 194 -> 193 -> 133 -> 36 -> 249 0 -> 213 -> 33 -> 3 -> 15 -> 249
54 -> 138	54 -> 44 -> 230 -> 138 54 -> 53 -> 60 -> 138 54 -> 55 -> 11 -> 12 -> 138 54 -> 153 -> 134 -> 95 -> 138 54 -> 78 -> 74 -> 238 -> 139 -> 138 54 -> 92 -> 17 -> 16 -> 15 -> 138 54 -> 185 -> 13 -> 201 -> 202 -> 138 54 -> 191 -> 63 -> 241 -> 137 -> 138
112 -> 5	112 -> 0 -> 7 -> 6 -> 5 112 -> 184 -> 185 -> 196 -> 5 112 -> 44 -> 45 -> 221 -> 4 -> 5 112 -> 111 -> 78 -> 190 -> 136 -> 5
45 -> 230	45 -> 44 -> 230 45 -> 46 -> 231 -> 230 45 -> 79 -> 208 -> 230 45 -> 90 -> 164 -> 230 45 -> 18 -> 53 -> 173 -> 230 45 -> 33 -> 21 -> 86 -> 230 45 -> 63 -> 161 -> 162 -> 230 45 -> 80 -> 81 -> 229 -> 230 45 -> 221 -> 222 -> 223 -> 230 45 -> 1 -> 2 -> 137 -> 138 -> 230 45 -> 144 -> 134 -> 83 -> 215 -> 230
0 -> 499	0 -> 101 -> 235 -> 499 0 -> 295 -> 296 -> 499 0 -> 385 -> 206 -> 499 0 -> 442 -> 443 -> 499 0 -> 1 -> 81 -> 55 -> 499 0 -> 70 -> 130 -> 498 -> 499 0 -> 199 -> 235 -> 246 -> 499 0 -> 269 -> 298 -> 169 -> 499 0 -> 342 -> 425 -> 392 -> 499

	0 -> 430 -> 15 -> 397 -> 499 0 -> 445 -> 435 -> 305 -> 499
357 -> 12	357 -> 12 357 -> 4 -> 470 -> 12 357 -> 36 -> 347 -> 12 357 -> 40 -> 130 -> 12 357 -> 87 -> 400 -> 12 357 -> 356 -> 487 -> 12 357 -> 358 -> 13 -> 12 357 -> 22 -> 138 -> 242 -> 12 357 -> 32 -> 31 -> 30 -> 12 357 -> 62 -> 481 -> 386 -> 12 357 -> 67 -> 68 -> 11 -> 12 357 -> 185 -> 94 -> 496 -> 12 357 -> 475 -> 474 -> 268 -> 12 357 -> 494 -> 161 -> 207 -> 12
25 -> 46	25 -> 24 -> 306 -> 46 25 -> 266 -> 279 -> 46 25 -> 433 -> 47 -> 46 25 -> 185 -> 359 -> 436 -> 46 25 -> 291 -> 213 -> 201 -> 46 25 -> 314 -> 313 -> 43 -> 46 25 -> 360 -> 280 -> 353 -> 46 25 -> 398 -> 313 -> 45 -> 46
130 -> 375	130 -> 79 -> 135 -> 375 130 -> 202 -> 299 -> 375 130 -> 12 -> 11 -> 237 -> 375 130 -> 26 -> 27 -> 441 -> 375 130 -> 40 -> 366 -> 252 -> 375 130 -> 70 -> 0 -> 402 -> 375 130 -> 129 -> 21 -> 194 -> 375 130 -> 131 -> 274 -> 376 -> 375 130 -> 283 -> 82 -> 220 -> 375 130 -> 341 -> 110 -> 374 -> 375 130 -> 351 -> 96 -> 211 -> 375 130 -> 352 -> 279 -> 278 -> 375 130 -> 498 -> 287 -> 117 -> 375 130 -> 67 -> 77 -> 196 -> 330 -> 375 130 -> 443 -> 154 -> 23 -> 471 -> 375

Tabela 2 - caminhos disjuntos encontrados pelo algoritmo

O Gráfico 1 mostra o resultado da eficiência do algoritmo implementado.

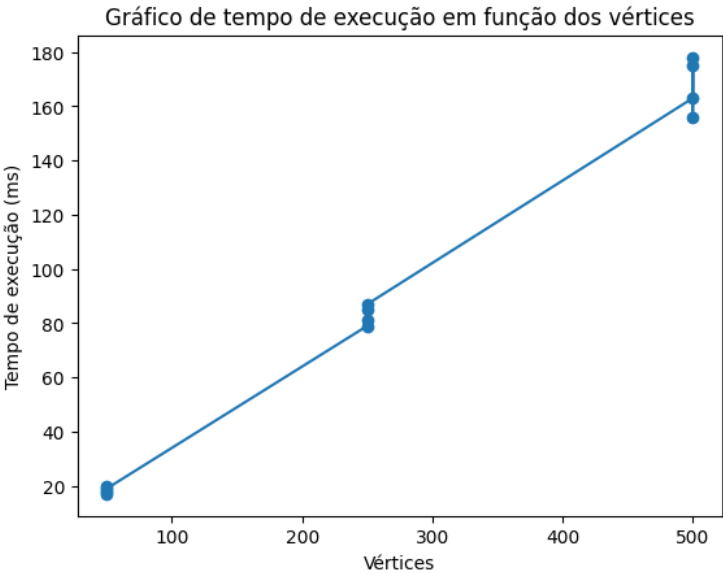


Gráfico 1 - tempo de execução do algoritmo em relação a quantidade de vértices do grafo