

Implementing a Breadth First Search Algorithm
Report #11

By
Eduardo Castro

CS 303 Algorithms and Data Structures

November 12, 2014

1. Problem Specification

The goal of this assignment is to implement a directed, an undirected graph and to implement a breadth first search algorithm on these graphs.

2. Program Design

The first part of the assignment was basically to extend a provided Graph class and implement an undirectedGraph and directedGraph classes, reads an input file to populate these graphs and print it in form of an adjacency list and check if it is right.

The following steps were required to develop this program:

- a) Create the Graph class
- b) Extend and write the undirectedGraph and directedGraph classes
- c) Use the driver class to display and measure the results

The following methods were defined within the Graph, UndirectedGraph and DirectedGraph:

- a) Graph()
 - a. Empty Constructor
- b) Graph(BufferedReader reader)
 - a. Build a graph with an input file
- c) addEdge(int v, int w)
 - a. adds an edge from element v to w. If it is a directed graph it points just from v to w and if it undirected it points from both sides
- d) toString()
 - a. prints the graph

Then we had to implement a BFS algorithm for an undirected graph. The following methods were defined for the BFS class:

- a) BFS(UndirectedGraph G, int s)
 - a. Starts the BFS and calls the bfs function
- b) bfs(UndirectedGraph G, int s)
 - a. Do the search based on source s
- c) printPath(UndirectedGraph, int s, int v)
 - a. print the path from element s to element v
- d) hasPathTo(int v)
 - a. returns true if there if a path to v
- e) pathTo(int v)
 - a. returns the path to v
- f) distTo(int v)
 - a. returns the dist to element v
- g) printPath(UndirectedGraph G)
 - a. print the path from the first element to the others

The Scanner class provided by Java was used to read in the necessary values within the provided driver program. The println method of the System.out object was used to display the inputs and results for the provided driver program.

3. Testing Plan

For the first part our test plan it is just to get the elements from a text file and prints the adjacency list. For the second part we will also read the text file for the input and print the path from the first element to the others. For the last part we need to measure and show the search times.

4. Test Cases

For the directed graph and input file tinyDG.txt

```
-----  
Directed Graph:  
There are 13 vertices and 22 edges  
0: 1 5  
1:  
2: 3 0  
3: 2 5  
4: 2 3  
5: 4  
6: 0 8 4 9  
7: 9 6  
8: 6  
9: 10 11  
10: 12  
11: 12 4  
12: 9  
-----
```

For the undirected graph and input file mediumG.txt

Undirected Graph:

There are 258 vertices and 5273 edges

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

134: 137 145 146 227 119 91 83 64 53 17
135: 241 181 230 200 86 71 42
136: 159 234 128 112 87 55
137: 145 146 183 215 218 224 227 134 119 109 91 64 47 29
138: 151 188 226 233 239 240 102 78 77 45 62 27 26 21 4
139: 156 196 205 210 214 219 221 122 110 100 101 92
140: 167 162 117 99 54 16 6
141: 116 94 86 42 35 18 2
142: 154 155 165 171 172 180 195 213 235 218 124 118 9
143: 152 175 179 193 212 244 246 131 106 105 30 51 8
144: 168 185 187 201 204 231 232 248 104 97 93 59 50 49 48 44 32
145: 146 137 134 120 119 91 73 64 56 53 34 22
146: 218 224 227 145 137 134 119 109 91 81 64 47 29
147: 162 166 140 129 117 99 54 16 6
148: 157 181 184 188 197 210 125 71 45 57 7
149: 163 206 209 213 222 225 80 46 39 24 15 0
150: 164 169 177 186 189 190 203 220 72 40 1
151: 168 187 200 226 231 138 118 77 65 57 52 32
152: 179 207 210 212 221 244 246 143 85 82 43 30 11 8
153: 228 241 115 95 76 37 31 3
154: 171 195 235 238 243 142 132
155: 165 171 172 180 197 213 235 238 142 124 118
156: 196 205 207 210 212 214 219 221 139 112 110 100 101 43 30
157: 181 184 188 197 210 146 125 101 71 65 57 42 7
158: 170 200 203 223 229 249 113 113 17
159: 214 239 136 120 112 70 55 4
160: 168 170 187 191 201 204 225 231 97 93 68 49 44 32 0
161: 169 177 186 189 229 249 120 56 46
162: 192 147 140 99 13
163: 202 209 211 222 225 149 114 58 33 24 15 0
164: 190 194 220 247 150 116 89 75 40 20 1
165: 171 172 180 191 213 155 142 124 118 68
166: 236 147 133 129 16 14 6
167: 224 117 47 29
168: 187 204 208 226 231 248 160 151 144 97 93 52 44 32
169: 177 186 189 190 220 161 150 46
170: 182 198 223 229 249 158 121 113 41 28 17 12
171: 172 180 213 216 218 245 165 155 154 142 132 124 92
172: 180 197 213 235 171 165 155 142 124 118 92 57
173: 128 107 90 69
174: 179 182 243 103 100 84 79 70 19 13
175: 246 143 129 85 82 11 10
176: 191 202 204 222 225 160 114 97 93 68 58 49 44 23 0
177: 180 189 203 249 169 161 150 72 46
178: 236 129 117 98 16 6
179: 193 212 244 174 152 143 133 106 84 79 70 30 19 8
180: 213 172 171 165 155 142 124 118
181: 184 188 196 230 157 148 135 125 100 101 71 65 57 42 7
182: 198 223 242 170 121 113 88 41 36 28 17 12
183: 215 137 126 109 74 64 38
184: 188 197 210 181 157 148 125 118 71 65 57 27 21 7
185: 201 232 244 184 184 97 93 59 50 49 48 44 32
186: 209 219 249 177 169 161 150 72 46
187: 200 226 231 248 168 151 144 182 93 77 52 32
188: 230 233 240 184 181 157 148 136 71 65 42 57 27 21 7
189: 200 203 220 249 186 177 169 161 150 72 1
190: 220 247 169 164 150 116 75 40 20
191: 202 204 222 225 231 176 165 160 114 97 93 68 58 49 44 0
192: 243 174 162 103 100 84 19 13
193: 243 179 143 131 106 105 84
194: 220 164 130 116 89 87 75 61 40 20 1
195: 218 241 154 142 23 9
196: 205 214 219 181 156 139 112 110 100 101 42
197: 210 184 172 157 155 148 126 124 118 45 57 7
198: 223 241 182 170 121 113 94 41 35 28 12
199: 217 96 63 40 25
200: 203 223 249 189 158 107 72 1
201: 216 217 232 248 185 144 104 83 50 48 32
202: 204 209 211 222 225 191 176 163 160 114 97 93 68 58 49 44 15 0
203: 249 200 189 186 177 158 150 107 72 1
204: 222 225 231 202 195 176 168 160 144 114 97 93 68 58 49 44 15 0
205: 207 218 214 219 215 196 156 139 122 110 100 101 92
206: 149 66 39 24
207: 218 212 214 219 221 244 205 156 152 122 110 82 43 30 11 8
208: 226 231 187 168 151 118 77 45 57 32 32
209: 211 212 215 200 202 163 149 114 66 58 39 24 15 0
210: 212 214 219 221 244 207 205 156 152 139 122 110 43 30 8
211: 222 225 209 202 163 149 114 66 58 39 24 15 0
212: 214 219 223 244 210 207 179 156 152 143 110 82 79 70 43 30 11 8
213: 235 238 180 172 171 165 155 142 124 118
214: 219 221 212 210 207 205 156 139 122 110 100 101 79 70 51 30
215: 183 157 116 109 74 64 38
216: 232 202 95 40
217: 232 203 204 83 67 55 48 45 26 5
218: 224 227 146 137 109 91 64 47 29
219: 221 214 212 210 207 205 156 139 122 110 100 101 43 30
220: 247 194 190 189 160 164 150 116 75 72 40 20 1
221: 219 214 212 210 207 205 156 152 139 122 110 43 30 8
222: 225 211 209 204 202 191 176 163 149 114 66 58 49 44 33 24 23 15 0
223: 242 249 200 198 182 170 158 121 113 18 17
224: 218 207 146 117 91 47 29
225: 222 211 209 204 202 191 176 163 160 149 114 97 93 68 58 49 44 24 15 0
226: 251 208 187 168 151 138 182 93 77 52 32 16 5
227: 218 146 137 114 119 91 81 64 29
228: 241 153 115 76 37 31 3
229: 249 170 162 158 120 81 53 17
230: 197 188 184 181 157 148 135 125 71 65 57 27 21 7
231: 248 216 200 204 191 187 168 160 151 144 97 93 52 44 32
232: 248 217 216 201 185 144 104 83 50 48 45
233: 240 200 110 90 72 42 27 21
234: 159 136 130 112 87 60
235: 230 213 172 171 155 154 142 132 124 92
236: 178 166 129 117 98 16 6
237: 199 96 63 60
238: 245 235 213 195 171 155 154 142 132
239: 240 239 138 128 112 78 62 51 26 4
240: 239 233 180 138 102 78 77 71 65 42 27 26 21 4
241: 218 153 115 76 37 31 3
242: 223 190 182 171 113 94 90 28 12
243: 193 192 174 131 106 105 103 84 19
244: 246 212 210 207 179 152 143 82 79 43 30 11 8
245: 210 195 171 154
246: 244 175 152 143 123 100 105 85 82 11 10 8
247: 220 190 164 116 75 40 20
248: 232 231 201 187 185 168 144 104 97 93 59 50 49 48 44 32
249: 229 223 203 200 189 186 177 170 161 154 72

Now the test results for the BFS Path. First for the smallG.txt input:

```
0 ----> 0:
0,
0 ----> 1:
1, 0,
0 ----> 2:
2, 1, 0,
0 ----> 3:
3, 0,
0 ----> 4:
4, 3, 0,
0 ----> 5:
5, 3, 0,
0 ----> 6:
6, 5, 3, 0,
0 ----> 7:
7, 5, 3, 0,
```

Now for the mediumG.txt input:

```
0 ---> 0:
0,
0 ---> 1:
1, 130, 234, 112, 78, 77, 187, 160, 0,
0 ---> 2:
2, 42, 157, 65, 208, 168, 44, 0,
0 ---> 3:
3, 45, 232, 144, 44, 0,
0 ---> 4:
4, 77, 187, 160, 0,
0 ---> 5:
5, 32, 160, 0,
0 ---> 6:
6, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 ---> 7:
7, 65, 208, 168, 44, 0,
0 ---> 8:
8, 207, 122, 92, 171, 165, 68, 0,
0 ---> 9:
9, 58, 0,
0 ---> 10:
10, 246, 244, 207, 122, 92, 171, 165, 68, 0,
0 ---> 11:
11, 207, 122, 92, 171, 165, 68, 0,
0 ---> 12:
12, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 13:
13, 174, 79, 214, 122, 92, 171, 165, 68, 0,
0 ---> 14:
14, 86, 135, 71, 65, 208, 168, 44, 0,
0 ---> 15:
15, 0,
0 ---> 16:
16, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 ---> 17:
17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 18:
18, 141, 135, 71, 65, 208, 168, 44, 0,
0 ---> 19:
19, 79, 214, 122, 92, 171, 165, 68, 0,
0 ---> 20:
20, 194, 87, 136, 55, 5, 32, 160, 0,
0 ---> 21:
21, 138, 226, 93, 44, 0,
0 ---> 22:
22, 120, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 23:
23, 58, 0,
0 ---> 24:
24, 0,
0 ---> 25:
25, 111, 87, 136, 55, 5, 32, 160, 0,
0 ---> 26:
26, 226, 93, 44, 0,
0 ---> 27:
27, 138, 226, 93, 44, 0,
0 ---> 28:
28, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 29:
29, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 30:
30, 156, 122, 92, 171, 165, 68, 0,
0 ---> 31:
31, 115, 95, 216, 201, 144, 44, 0,
0 ---> 32:
32, 160, 0,
0 ---> 33:
33, 58, 0,
0 ---> 34:
34, 120, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 35:
35, 141, 135, 71, 65, 208, 168, 44, 0,
0 ---> 36:
36, 182, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 37:
37, 95, 216, 201, 144, 44, 0,
0 ---> 38:
38, 109, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 39:
39, 15, 0,
0 ---> 40:
40, 194, 87, 136, 55, 5, 32, 160, 0,
0 ---> 41:
41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 42:
42, 157, 65, 208, 168, 44, 0,
0 ---> 43:
43, 156, 122, 92, 171, 165, 68, 0,
0 ---> 44:
44, 0,
0 ---> 45:
45, 232, 144, 44, 0,
0 ---> 46:
46, 186, 203, 107, 173, 128, 78, 77, 187, 160, 0,
0 ---> 47:
47, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 48:
48, 144, 44, 0,
0 ---> 49:
49, 0,
0 ---> 50:
50, 59, 0,
0 ---> 51:
51, 214, 122, 92, 171, 165, 68, 0,
0 ---> 52:
52, 93, 44, 0,
0 ---> 53:
53, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 54:
54, 147, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 ---> 55:
55, 5, 32, 160, 0,
0 ---> 56:
56, 161, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 57:
57, 208, 168, 44, 0,
0 ---> 58:
58, 0,
0 ---> 59:
59, 0,
0 ---> 60:
60, 111, 87, 136, 55, 5, 32, 160, 0,
0 ---> 61:
61, 234, 112, 78, 77, 187, 160, 0,
0 ---> 62:
62, 138, 226, 93, 44, 0,
0 ---> 63:
63, 60, 111, 87, 136, 55, 5, 32, 160, 0,
0 ---> 64:
64, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 65:
65, 208, 168, 44, 0,
0 ---> 66:
66, 15, 0,
0 ---> 67:
67, 5, 32, 160, 0,
```

```

0 --> 68:
68, 0,
0 --> 69:
69, 128, 78, 77, 187, 160, 0,
0 --> 70:
70, 214, 122, 92, 171, 165, 68, 0,
0 --> 71:
71, 65, 208, 168, 44, 0,
0 --> 72:
72, 187, 173, 128, 78, 77, 187, 160, 0,
0 --> 73:
73, 120, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 74:
74, 189, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 75:
75, 134, 87, 136, 55, 5, 32, 160, 0,
0 --> 76:
76, 45, 232, 144, 44, 0,
0 --> 77:
77, 187, 160, 0,
0 --> 78:
78, 77, 187, 160, 0,
0 --> 79:
79, 214, 122, 92, 171, 165, 68, 0,
0 --> 80:
80, 0,
0 --> 81:
81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 82:
82, 207, 122, 92, 171, 165, 68, 0,
0 --> 83:
83, 201, 144, 44, 0,
0 --> 84:
84, 79, 214, 122, 92, 171, 165, 68, 0,
0 --> 85:
85, 152, 207, 122, 92, 171, 165, 68, 0,
0 --> 86:
86, 135, 71, 65, 208, 168, 44, 0,
0 --> 87:
87, 136, 55, 5, 32, 160, 0,
0 --> 88:
88, 182, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 89:
89, 130, 234, 112, 78, 77, 187, 160, 0,
0 --> 90:
90, 233, 138, 226, 93, 44, 0,
0 --> 91:
91, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 92:
92, 171, 165, 68, 0,
0 --> 93:
93, 44, 0,
0 --> 94:
94, 242, 90, 233, 138, 226, 93, 44, 0,
0 --> 95:
95, 216, 201, 144, 44, 0,
0 --> 96:
96, 60, 111, 87, 136, 55, 5, 32, 160, 0,
0 --> 97:
97, 0,
0 --> 98:
98, 88, 182, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 99:
99, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 --> 100:
100, 70, 214, 122, 92, 171, 165, 68, 0,
0 --> 101:
101, 125, 172, 165, 68, 0,
0 --> 102:
102, 187, 160, 0,
0 --> 103:
103, 84, 79, 214, 122, 92, 171, 165, 68, 0,
0 --> 104:
104, 144, 44, 0,
0 --> 105:
105, 143, 212, 156, 122, 92, 171, 165, 68, 0,
0 --> 106:
106, 179, 212, 156, 122, 92, 171, 165, 68, 0,
0 --> 107:
107, 173, 128, 78, 77, 187, 160, 0,
0 --> 108:
108, 181, 65, 208, 168, 44, 0,
0 --> 109:
109, 140, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 110:
110, 122, 92, 171, 165, 68, 0,
0 --> 111:
111, 87, 136, 55, 5, 32, 160, 0,
0 --> 112:
112, 78, 77, 187, 160, 0,
0 --> 113:
113, 90, 233, 138, 226, 93, 44, 0,
0 --> 114:
114, 0,
0 --> 115:
115, 95, 216, 201, 144, 44, 0,
0 --> 116:
116, 194, 87, 136, 55, 5, 32, 160, 0,
0 --> 117:
117, 98, 88, 182, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 118:
118, 165, 68, 0,
0 --> 119:
119, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 120:
120, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 121:
121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 122:
122, 92, 171, 165, 68, 0,
0 --> 123:
123, 246, 244, 207, 122, 92, 171, 165, 68, 0,
0 --> 124:
124, 165, 68, 0,
0 --> 125:
125, 172, 165, 68, 0,
0 --> 126:
126, 100, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 --> 127:
127, 89, 130, 234, 112, 78, 77, 187, 160, 0,
0 --> 128:
128, 78, 77, 187, 160, 0,
0 --> 129:
129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 --> 130:
130, 234, 112, 78, 77, 187, 160, 0,
0 --> 131:
131, 179, 212, 156, 122, 92, 171, 165, 68, 0,
0 --> 132:
132, 171, 165, 68, 0,
0 --> 133:
133, 51, 214, 122, 92, 171, 165, 68, 0,
0 --> 134:
134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,

```


0 ---> 135:
 135, 71, 65, 208, 168, 44, 0,
 0 ---> 136:
 136, 55, 5, 32, 160, 0,
 0 ---> 137:
 137, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 138:
 138, 226, 93, 44, 0,
 0 ---> 139:
 139, 92, 171, 165, 68, 0,
 0 ---> 140:
 140, 147, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
 0 ---> 141:
 141, 135, 71, 65, 208, 168, 44, 0,
 0 ---> 142:
 142, 9, 58, 0,
 0 ---> 143:
 143, 212, 156, 122, 92, 171, 165, 68, 0,
 0 ---> 144:
 144, 44, 0,
 0 ---> 145:
 145, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 146:
 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 147:
 147, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
 0 ---> 148:
 148, 65, 208, 168, 44, 0,
 0 ---> 149:
 149, 0,
 0 ---> 150:
 150, 1, 130, 234, 112, 78, 77, 187, 160, 0,
 0 ---> 151:
 151, 168, 44, 0,
 0 ---> 152:
 152, 207, 122, 92, 171, 165, 68, 0,
 0 ---> 153:
 153, 95, 216, 201, 144, 44, 0,
 0 ---> 154:
 154, 195, 23, 58, 0,
 0 ---> 155:
 155, 165, 68, 0,
 0 ---> 156:
 156, 122, 92, 171, 165, 68, 0,
 0 ---> 157:
 157, 65, 208, 168, 44, 0,
 0 ---> 158:
 158, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 159:
 159, 78, 77, 187, 160, 0,
 0 ---> 160:
 160, 0,
 0 ---> 161:
 161, 229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 162:
 162, 192, 84, 79, 214, 122, 92, 171, 165, 68, 0,
 0 ---> 163:
 163, 0,
 0 ---> 164:
 164, 194, 87, 136, 55, 5, 32, 160, 0,
 0 ---> 165:
 165, 68, 0,
 0 ---> 166:
 166, 14, 86, 135, 71, 65, 208, 168, 44, 0,
 0 ---> 167:
 167, 117, 98, 88, 182, 113, 90, 233, 138, 226, 93, 44, 0,

0 ---> 168:
 168, 44, 0,
 0 ---> 169:
 169, 220, 194, 87, 136, 55, 5, 32, 160, 0,
 0 ---> 170:
 170, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 171:
 171, 165, 68, 0,
 0 ---> 172:
 172, 165, 68, 0,
 0 ---> 173:
 173, 128, 78, 77, 187, 160, 0,
 0 ---> 174:
 174, 79, 214, 122, 92, 171, 165, 68, 0,
 0 ---> 175:
 175, 82, 207, 122, 92, 171, 165, 68, 0,
 0 ---> 176:
 176, 0,
 0 ---> 177:
 177, 203, 107, 173, 128, 78, 77, 187, 160, 0,
 0 ---> 178:
 178, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
 0 ---> 179:
 179, 212, 156, 122, 92, 171, 165, 68, 0,
 0 ---> 180:
 180, 165, 68, 0,
 0 ---> 181:
 181, 65, 208, 168, 44, 0,
 0 ---> 182:
 182, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 183:
 183, 137, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 184:
 184, 118, 165, 68, 0,
 0 ---> 185:
 185, 44, 0,
 0 ---> 186:
 186, 203, 107, 173, 128, 78, 77, 187, 160, 0,
 0 ---> 187:
 187, 160, 0,
 0 ---> 188:
 188, 138, 226, 93, 44, 0,
 0 ---> 189:
 189, 1, 130, 234, 112, 78, 77, 187, 160, 0,
 0 ---> 190:
 190, 220, 194, 87, 136, 55, 5, 32, 160, 0,
 0 ---> 191:
 191, 0,
 0 ---> 192:
 192, 84, 79, 214, 122, 92, 171, 165, 68, 0,
 0 ---> 193:
 193, 179, 212, 156, 122, 92, 171, 165, 68, 0,
 0 ---> 194:
 194, 87, 136, 55, 5, 32, 160, 0,
 0 ---> 195:
 195, 23, 58, 0,
 0 ---> 196:
 196, 181, 65, 208, 168, 44, 0,
 0 ---> 197:
 197, 172, 165, 68, 0,
 0 ---> 198:
 198, 113, 90, 233, 138, 226, 93, 44, 0,
 0 ---> 199:
 199, 60, 111, 87, 136, 55, 5, 32, 160, 0,
 0 ---> 200:
 200, 107, 173, 128, 78, 77, 187, 160, 0,


```

0 ---> 202:
202, 0,
0 ---> 203:
203, 107, 173, 128, 78, 77, 187, 160, 0,
0 ---> 204:
204, 0,
0 ---> 205:
205, 92, 171, 165, 68, 0,
0 ---> 206:
206, 24, 0,
0 ---> 207:
207, 122, 92, 171, 165, 68, 0,
0 ---> 208:
208, 168, 44, 0,
0 ---> 209:
209, 0,
0 ---> 210:
210, 122, 92, 171, 165, 68, 0,
0 ---> 211:
211, 0,
0 ---> 212:
212, 156, 122, 92, 171, 165, 68, 0,
0 ---> 213:
213, 165, 68, 0,
0 ---> 214:
214, 122, 92, 171, 165, 68, 0,
0 ---> 215:
215, 137, 134, 17, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 216:
216, 201, 144, 44, 0,
0 ---> 217:
217, 201, 144, 44, 0,
0 ---> 218:
218, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 219:
219, 122, 92, 171, 165, 68, 0,
0 ---> 220:
220, 194, 87, 136, 55, 5, 32, 160, 0,
0 ---> 221:
221, 122, 92, 171, 165, 68, 0,
0 ---> 222:
222, 0,
0 ---> 223:
223, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 224:
224, 146, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 225:
225, 0,
0 ---> 226:
226, 93, 44, 0,
0 ---> 227:
227, 81, 41, 121, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 228:
228, 76, 45, 232, 144, 44, 0,
0 ---> 229:
229, 158, 113, 90, 233, 138, 226, 93, 44, 0,
0 ---> 230:
230, 65, 208, 168, 44, 0,
0 ---> 231:
231, 44, 0,
0 ---> 232:
232, 144, 44, 0,
0 ---> 233:
233, 138, 226, 93, 44, 0,
0 ---> 234:
234, 112, 78, 77, 187, 160, 0,

```

```

0 ---> 235:
235, 142, 9, 58, 0,
0 ---> 236:
236, 129, 14, 86, 135, 71, 65, 208, 168, 44, 0,
0 ---> 237:
237, 60, 111, 87, 136, 55, 5, 32, 160, 0,
0 ---> 238:
238, 195, 23, 58, 0,
0 ---> 239:
239, 138, 226, 93, 44, 0,
0 ---> 240:
240, 102, 187, 160, 0,
0 ---> 241:
241, 76, 45, 232, 144, 44, 0,
0 ---> 242:
242, 90, 233, 138, 226, 93, 44, 0,
0 ---> 243:
243, 84, 79, 214, 122, 92, 171, 165, 68, 0,
0 ---> 244:
244, 207, 122, 92, 171, 165, 68, 0,
0 ---> 245:
245, 195, 23, 58, 0,
0 ---> 246:
246, 244, 207, 122, 92, 171, 165, 68, 0,
0 ---> 247:
247, 220, 194, 87, 136, 55, 5, 32, 160, 0,
0 ---> 248:
248, 44, 0,
0 ---> 249:
249, 158, 113, 90, 233, 138, 226, 93, 44, 0,

```

Now the adjacency matrix got from the smallG.txt input file

```
      0  1  2  3  4  5  6  7
0:  0  1  0  1  0  0  0  0
1:  1  0  1  0  0  0  0  0
2:  0  1  0  0  0  0  0  0
3:  1  0  0  0  1  1  0  0
4:  0  0  0  1  0  1  1  0
5:  0  0  0  1  1  0  1  1
6:  0  0  0  0  1  1  0  1
7:  0  0  0  0  0  1  1  0
```

Now according to the measured times I have the following times:

	largeG.txt	mediumG.txt	tinyDG.txt
Adjacency List		19192866	3261414
Adjacency Matrix			

4. Conclusion

Instead implementing the undirected and directed graph were pretty easy the breadth first search was quite difficult. For the graphs there were no secret: with the provided graph class provided we had just to implement the addEdge function according to the specification of each

kind of graph. Even using adjacency matrix instead of adjacency list was not a big deal, only a small change in the class was necessary.

For the Breadth First Search we can consider that the time complexity is $O(bd)$, with b representing the branching factor of the graph (number of children of each vertex) and d representing the depth of the graph. The BFS considers all possible paths to all nodes. This algorithm is a great way to discover the smallest way from element X to element Y and can be very useful for some cases.

I couldn't use it with adjacency matrix, I implemented part of it as I believe that it is necessary but it is not working. Even for the adjacency list I had a lot of problems related to `NullPointerException` (probably related to the `Vertex` structure) but after some modifications I made it work.

As the `largeG.txt` file was 99MB I also couldn't download it since my browser keeps crashing every time that I try to copy it because of its size.