



src/Tests/PartA\_DFSTest.java

```
1 packageimportimportimportstaticimportimportimportimportpublicclassPartA_DFSTest
privatefinalByteArrayOutputStreamoutContent=newByteArrayOutputStream@Test
publicvoidtestBasicPathfinding()newPrintStreamNodegoal=newNode1180null0null
Nodestart=newNode190null0intplanetSize=4Stringfrontier_result="[(1:90)]\n"
"[(1:45),(1:135),(2:90)]\n" "["(1:0),(2:45),(1:135),(2:90)]\n" "["(1:315),(2:0),
(2:45),(1:135),(2:90)]\n" "["(1:270),(2:315),(2:0),(2:45),(1:135),(2:90)]\n"
"[(1:225),(2:270),(2:315),(2:0),(2:45),(1:135),(2:90)]\n" "["(1:180),(2:225),
(2:270),(2:315),(2:0),(2:45),(1:135),(2:90)]\n" "(1:90)(1:45)(1:0)(1:315)
(1:270)(1:225)(1:180)\n" "4.712\n" "7\n"@TestpublicvoidtestAdvancedPathfinding()
newPrintStreamNodegoal=newNode290null0nullNodestart=newNode70null0int
planetSize=8Stringfrontier_result="[(7:0)]\n" "["(6:0),(7:45),(7:315)]\n"
"[(5:0),(6:45),(6:315),(7:45),(7:315)]\n" "["(4:0),(5:45),(5:315),(6:45),
(6:315),(7:45),(7:315)]\n" "["(3:0),(4:45),(4:315),(5:45),(5:315),(6:45),
(6:315),(7:45),(7:315)]\n" "["(2:0),(3:45),(3:315),(4:45),(4:315),(5:45),
(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(1:0),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(1:45),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),
(6:45),(6:315),(7:45),(7:315)]\n" "["(1:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(1:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),
(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(1:180),(2:135),(2:90),(1:315),
(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),
(7:45),(7:315)]\n" "["(1:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),
(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(1:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(2:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(3:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(3:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),
(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(3:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),
(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),
(7:315)]\n" "["(3:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),
(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),
(6:315),(7:45),(7:315)]\n" "["(3:90),(4:135),(4:180),(4:225),(4:270),(2:225),
(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),
(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(4:90),(4:135),(4:180),
(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(5:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),
(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),
(6:315),(7:45),(7:315)]\n" "["(5:135),(6:90),(4:135),(4:180),(4:225),(4:270),
(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),
(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(5:180),(6:135),
(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),
(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),
(7:45),(7:315)]\n" "["(5:225),(6:180),(6:135),(6:90),(4:135),(4:180),(4:225),
(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),
(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(5:270),
(6:225),(6:180),(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),
(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),
(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n" "["(6:270),(6:225),(6:180),
(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),
(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),
(6:315),(7:45),(7:315)]\n" "["(7:270),(6:225),(6:180),(6:135),(6:90),(4:135),
(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),
```

```

(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(7:225),(6:225),(6:180),(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),
(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),
(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(7:180),(6:225),
(6:180),(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),
(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),
(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(7:135),(6:225),(6:180),(6:135),
(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),
(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),
(7:45),(7:315)]\n"[(7:90),(6:225),(6:180),(6:135),(6:90),(4:135),(4:180),
(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),
(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(6:225),(6:180),(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),
(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),
(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(6:180),(6:135),(6:90),
(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),
(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),
(7:315)]\n"[(6:135),(6:90),(4:135),(4:180),(4:225),(4:270),(2:225),(2:180),
(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),
(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(6:90),(4:135),(4:180),(4:225),
(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),
(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(4:135),
(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),
(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(4:180),(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),
(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),
(7:315)]\n"[(4:225),(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),
(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),
(7:315)]\n"[(4:270),(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),
(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(2:225),(2:180),(2:135),(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),
(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"[(2:180),(2:135),
(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),
(6:45),(6:315),(7:45),(7:315)]\n"[(2:135),(2:90),(1:315),(2:45),(2:315),
(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),(6:45),(6:315),(7:45),(7:315)]\n"
"[(2:90),(1:315),(2:45),(2:315),(3:45),(3:315),(4:45),(4:315),(5:45),(5:315),
(6:45),(6:315),(7:45),(7:315)]\n"(7:0)(6:0)(5:0)(4:0)(3:0)(2:0)(1:0)(1:45)
(1:90)(2:90)\n"8.571\n"43\n"@TestpublicvoidtestAdvancedPathfinding2()new
PrintStreamNodegoal=newNode10null0nullNodestart=newNode3180null0intplanetSize=
4Stringfrontier_result="[(3:180)]\n"[(2:180),(3:135),(3:225)]\n"[(1:180),
(2:135),(2:225),(3:135),(3:225)]\n"[(1:135),(1:225),(2:135),(2:225),(3:135),
(3:225)]\n"[(1:90),(1:225),(2:135),(2:225),(3:135),(3:225)]\n"[(1:45),
(2:90),(1:225),(2:135),(2:225),(3:135),(3:225)]\n"[(1:0),(2:45),(2:90),
(1:225),(2:135),(2:225),(3:135),(3:225)]\n"[(3:180)(2:180)(1:180)(1:135)(1:90)
(1:45)(1:0)\n"5.142\n"7\n"@TestpublicvoidtestGoalOfZero()newPrintStreamNode
goal=newNode00null0nullNodestart=newNode10null0intplanetSize=2String
frontier_result="[(1:0)]\n"[(1:45),(1:315)]\n"[(1:90),(1:315)]\n"[(1:135),
(1:315)]\n"[(1:180),(1:315)]\n"[(1:225),(1:315)]\n"[(1:270),(1:315)]\n"
"[(1:315)]\n"fail\n"8\n"// expected here is weirdly long?@Testpublicvoid
testStartOfZero()newPrintStreamNodegoal=newNode20null0nullNodestart=newNode00
null0intplanetSize=2Stringfrontier_result="[(0:0)]\n"[(1:0)]\n"[(1:45),
(1:315)]\n"[(1:90),(1:315)]\n"[(1:135),(1:315)]\n"[(1:180),(1:315)]\n"
"[(1:225),(1:315)]\n"[(1:270),(1:315)]\n"[(1:315)]\n"fail\n"9\n"@Test
publicvoidtestEdgeOfTheGrid()newPrintStreamNodegoal=newNode345null0nullNode
start=newNode30null0intplanetSize=2Stringfrontier_result="[(3:0)]\n"fail\n"
1\n"@TestpublicvoidtestGoalOffGrid()newPrintStreamNodegoal=newNode3315null0
nullNodestart=newNode1270null0intplanetSize=2Stringfrontier_result=
"[(1:270)]\n"[(1:225),(1:315)]\n"[(1:180),(1:315)]\n"[(1:135),(1:315)]\n"
"[(1:90),(1:315)]\n"[(1:45),(1:315)]\n"[(1:0),(1:315)]\n"[(1:315)]\n"
"fail\n"8\n"@TestpublicvoidtestStartOffGrid()newPrintStreamNodegoal=newNode1
315null0nullNodestart=newNode4270null0intplanetSize=2Stringfrontier_result=

```

```

"[(4:270)]\n"fail\n"1\n" Tests;
2
3  General.Node;
4  Algorithms.PartA_DFS;
5    org.junit.Assert.*;
6
7  java.io.ByteArrayOutputStream;
8  java.io.PrintStream;
9
10 org.junit.Test;
11
12 {
13
14     ();
15
16
17     {
18         System.setOut( (outContent));
19
20         ( , , , , );
21         ( , , , , goal);
22         ;
23
24         +
25         +
26         +
27         +
28         +
29         +
30         +
31         +
32         + ;
33
34         PartA_DFS.dfs(start, goal, planetSize);
35         assertEquals(frontier_result, outContent.toString());
36     }
37
38
39     {
40         System.setOut( (outContent));
41
42         ( , , , , );
43         ( , , , , goal);
44         ;
45
46         +
47         +
48         +
49         +
50         +
51         +
52         +
53         +
54         +
55         +
56         +
57         +
58         +
59         +
60         +
61         +

```

```

62         +
63         +
64         +
65         +
66         +
67         +
68         +
69         +
70         +
71         +
72         +
73         +
74         +
75         +
76         +
77         +
78         +
79         +
80         +
81         +
82         +
83         +
84         +
85         +
86         +
87         +
88         +
89         +
90         + ;
91
92     PartA_DFS.dfs(start, goal, planetSize);
93     assertEquals(frontier_result, outContent.toString());
94 }
95
96
97 {
98     System.setOut( (outContent));
99
100         ( , , , , );
101         ( , , , , goal);
102     ;
103
104         +
105         +
106         +
107         +
108         +
109         +
110         +
111         +
112         + ;
113
114     PartA_DFS.dfs(start, goal, planetSize);
115     assertEquals(frontier_result, outContent.toString());
116 }
117
118
119 {
120     System.setOut( (outContent));
121
122         ( , , , , );

```

```

123         ( , , , , goal);
124     ;
125
126         +
127         +
128         +
129         +
130         +
131         +
132         +
133         +
134         + ;
135
136     PartA_DFS.dfs(start, goal, planetSize);
137     assertEquals(frontier_result, outContent.toString());
138 }
139
140
141 {
142     System.setOut( (outContent));
143
144         ( , , , , );
145         ( , , , , goal);
146     ;
147
148         +
149         +
150         +
151         +
152         +
153         +
154         +
155         +
156         +
157         + ;
158
159     PartA_DFS.dfs(start, goal, planetSize);
160     assertEquals(frontier_result, outContent.toString());
161 }
162
163
164 {
165     System.setOut( (outContent));
166
167         ( , , , , );
168         ( , , , , goal);
169     ;
170
171         +
172         + ;
173
174     PartA_DFS.dfs(start, goal, planetSize);
175     assertEquals(frontier_result, outContent.toString());
176 }
177
178
179 {
180     System.setOut( (outContent));
181
182         ( , , , , );
183         ( , , , , goal);

```

```

184         ;
185
186         +
187         +
188         +
189         +
190         +
191         +
192         +
193         +
194         + ;
195
196     PartA_DFS.dfs(start, goal, planetSize);
197     assertEquals(frontier_result, outContent.toString());
198 }
199
200
201 {
202     System.setOut( (outContent));
203
204     ( , , , , );
205     ( , , , , goal);
206     ;
207
208     +
209     + ;
210
211     PartA_DFS.dfs(start, goal, planetSize);
212     assertEquals(frontier_result, outContent.toString());
213 }
214 }
215

```