src/Tests/PartA_BFSTest.java

```
  1 packageimportimportimportstaticimportimportimportpublicclassPartA_BFSTest
privatefinalByteArrayOutputStreamoutContent=newByteArrayOutputStream@Test
publicvoidtestBasicPathfinding()newPrintStreamNodegoal=newNode390null0nullNode
start=newNode30null0intplanetSize=4Stringfrontier_result="[(3:0)]\n""[(2:0),
(3:45),(3:315)]\n""[(3:45),(3:315),(1:0),(2:45),(2:315)]\n""[(3:315),(1:0),
(2:45),(2:315),(3:90)]\n""[(1:0),(2:45),(2:315),(3:90),(3:270)]\n""[(2:45),
(2:315),(3:90),(3:270),(1:45),(1:315)]\n""[(2:315),(3:90),(3:270),(1:45),
(1:315),(2:90)]\n""[(3:90),(3:270),(1:45),(1:315),(2:90),(2:270)]\n""(3:0)
(3:45)(3:90)\n""4.712\n""8\n"@Testpublicvoidtestadvancedpathfinding()new
PrintStreamNodegoal=newNode290null0nullNodestart=newNode70null0intplanetSize=8
Stringfrontier_result="[(7:0)]\n""[(6:0),(7:45),(7:315)]\n""[(7:45),(7:315),
(5:0),(6:45),(6:315)]\n""[(7:315),(5:0),(6:45),(6:315),(7:90)]\n""[(5:0),
(6:45),(6:315),(7:90),(7:270)]\n""[(6:45),(6:315),(7:90),(7:270),(4:0),(5:45),
(5:315)]\n""[(6:315),(7:90),(7:270),(4:0),(5:45),(5:315),(6:90)]\n""[(7:90),
(7:270),(4:0),(5:45),(5:315),(6:90),(6:270)]\n""[(7:270),(4:0),(5:45),(5:315),
(6:90),(6:270),(7:135)]\n""[(4:0),(5:45),(5:315),(6:90),(6:270),(7:135),
(7:225)]\n""[(5:45),(5:315),(6:90),(6:270),(7:135),(7:225),(3:0),(4:45),
(4:315)]\n""[(5:315),(6:90),(6:270),(7:135),(7:225),(3:0),(4:45),(4:315),
(5:90)]\n""[(6:90),(6:270),(7:135),(7:225),(3:0),(4:45),(4:315),(5:90),
(5:270)]\n""[(6:270),(7:135),(7:225),(3:0),(4:45),(4:315),(5:90),(5:270),
(6:135)]\n""[(7:135),(7:225),(3:0),(4:45),(4:315),(5:90),(5:270),(6:135),
(6:225)]\n""[(7:225),(3:0),(4:45),(4:315),(5:90),(5:270),(6:135),(6:225),
(7:180)]\n""[(3:0),(4:45),(4:315),(5:90),(5:270),(6:135),(6:225),(7:180)]\n"
"[(4:45),(4:315),(5:90),(5:270),(6:135),(6:225),(7:180),(2:0),(3:45),
(3:315)]\n""[(4:315),(5:90),(5:270),(6:135),(6:225),(7:180),(2:0),(3:45),
(3:315),(4:90)]\n""[(5:90),(5:270),(6:135),(6:225),(7:180),(2:0),(3:45),
(3:315),(4:90),(4:270)]\n""[(5:270),(6:135),(6:225),(7:180),(2:0),(3:45),
(3:315),(4:90),(4:270),(5:135)]\n""[(6:135),(6:225),(7:180),(2:0),(3:45),
(3:315),(4:90),(4:270),(5:135),(5:225)]\n""[(6:225),(7:180),(2:0),(3:45),
(3:315),(4:90),(4:270),(5:135),(5:225),(6:180)]\n""[(7:180),(2:0),(3:45),
(3:315),(4:90),(4:270),(5:135),(5:225),(6:180)]\n""[(2:0),(3:45),(3:315),
(4:90),(4:270),(5:135),(5:225),(6:180)]\n""[(3:45),(3:315),(4:90),(4:270),
(5:135),(5:225),(6:180),(1:0),(2:45),(2:315)]\n""[(3:315),(4:90),(4:270),
(5:135),(5:225),(6:180),(1:0),(2:45),(2:315),(3:90)]\n""[(4:90),(4:270),
(5:135),(5:225),(6:180),(1:0),(2:45),(2:315),(3:90),(3:270)]\n""[(4:270),
(5:135),(5:225),(6:180),(1:0),(2:45),(2:315),(3:90),(3:270),(4:135)]\n"
"[(5:135),(5:225),(6:180),(1:0),(2:45),(2:315),(3:90),(3:270),(4:135),
(4:225)]\n""[(5:225),(6:180),(1:0),(2:45),(2:315),(3:90),(3:270),(4:135),
(4:225),(5:180)]\n""[(6:180),(1:0),(2:45),(2:315),(3:90),(3:270),(4:135),
(4:225),(5:180)]\n""[(1:0),(2:45),(2:315),(3:90),(3:270),(4:135),(4:225),
(5:180)]\n""[(2:45),(2:315),(3:90),(3:270),(4:135),(4:225),(5:180),(1:45),
(1:315)]\n""[(2:315),(3:90),(3:270),(4:135),(4:225),(5:180),(1:45),(1:315),
(2:90)]\n""[(3:90),(3:270),(4:135),(4:225),(5:180),(1:45),(1:315),(2:90),
(2:270)]\n""[(3:270),(4:135),(4:225),(5:180),(1:45),(1:315),(2:90),(2:270),
(3:135)]\n""[(4:135),(4:225),(5:180),(1:45),(1:315),(2:90),(2:270),(3:135),
(3:225)]\n""[(4:225),(5:180),(1:45),(1:315),(2:90),(2:270),(3:135),(3:225),
(4:180)]\n""[(5:180),(1:45),(1:315),(2:90),(2:270),(3:135),(3:225),(4:180)]\n"
"[(1:45),(1:315),(2:90),(2:270),(3:135),(3:225),(4:180)]\n""[(1:315),(2:90),
(2:270),(3:135),(3:225),(4:180),(1:90)]\n""[(2:90),(2:270),(3:135),(3:225),
(4:180),(1:90),(1:270)]\n""(7:0)(6:0)(5:0)(4:0)(3:0)(2:0)(2:45)(2:90)\n"
"8.142\n""43\n"@Testpublicvoidtestadvancedpathfinding2()newPrintStreamNodegoal
=newNode10null0nullNodestart=newNode3180null0intplanetSize=4String
frontier_result="[(3:180)]\n""[(2:180),(3:135),(3:225)]\n""[(3:135),(3:225),
(1:180),(2:135),(2:225)]\n""[(3:225),(1:180),(2:135),(2:225),(3:90)]\n"
"[(1:180),(2:135),(2:225),(3:90),(3:270)]\n""[(2:135),(2:225),(3:90),(3:270),
(1:135),(1:225)]\n""[(2:225),(3:90),(3:270),(1:135),(1:225),(2:90)]\n"
"[(3:90),(3:270),(1:135),(1:225),(2:90),(2:270)]\n""[(3:270),(1:135),(1:225),
(2:90),(2:270),(3:45)]\n""[(1:135),(1:225),(2:90),(2:270),(3:45),(3:315)]\n"
```

```
"[(1:225),(2:90),(2:270),(3:45),(3:315),(1:90)]\n""[(2:90),(2:270),(3:45),
(3:315),(1:90),(1:270)]\n""[(2:270),(3:45),(3:315),(1:90),(1:270),(2:45)]\n"
"[(3:45),(3:315),(1:90),(1:270),(2:45),(2:315)]\n""[(3:315),(1:90),(1:270),
(2:45),(2:315),(3:0)]\n""[(1:90),(1:270),(2:45),(2:315),(3:0)]\n""[(1:270),
(2:45),(2:315),(3:0),(1:45)]\n""[(2:45),(2:315),(3:0),(1:45),(1:315)]\n"
"[(2:315),(3:0),(1:45),(1:315),(2:0)]\n""[(3:0),(1:45),(1:315),(2:0)]\n"
"[(1:45),(1:315),(2:0)]\n""[(1:315),(2:0),(1:0)]\n""[(2:0),(1:0)]\n"
"[(1:0)]\n""(3:180)(2:180)(1:180)(1:135)(1:90)(1:45)(1:0)\n""5.142\n""24\n"
@Testpublicvoidtest GoalOfZero()newPrintStreamNodegoal=newNode00null0nullNode
start=newNode10null0intplanetSize=2Stringfrontier_result="[(1:0)]\n""[(1:45),
(1:315)]\n""[(1:315),(1:90)]\n""[(1:90),(1:270)]\n""[(1:270),(1:135)]\n"
"[(1:135),(1:225)]\n""[(1:225),(1:180)]\n""[(1:180)]\n""fail\n""8\n"@Test
publicvoidtestStartOfZero()newPrintStreamNodegoal=newNode20null0nullNodestart=
newNode00null0intplanetSize=5Stringfrontier_result="[(0:0)]\n""[(1:0)]\n"
"[(1:45),(1:315),(2:0)]\n""[(1:315),(2:0),(1:90),(2:45)]\n""[(2:0),(1:90),
(2:45),(1:270),(2:315)]\n""(0:0)(1:0)(2:0)\n""2.000\n""5\n"@Testpublicvoid
testEdgeOfTheGrid()newPrintStreamNodegoal=newNode345null0nullNodestart=newNode
30null0intplanetSize=2Stringfrontier_result="[(3:0)]\n""fail\n""1\n"@Test
publicvoidtestGoalOffGrid()newPrintStreamNodegoal=newNode3315null0nullNode
start=newNode1270null0intplanetSize=2Stringfrontier_result="[(1:270)]\n"
"[(1:225),(1:315)]\n""[(1:315),(1:180)]\n""[(1:180),(1:0)]\n""[(1:0),
(1:135)]\n""[(1:135),(1:45)]\n""[(1:45),(1:90)]\n""[(1:90)]\n""fail\n""8\n"
@Testpublicvoidtest StartOffGrid()newPrintStreamNodegoal=newNode1315null0null
Nodestart=newNode4270null0intplanetSize=2Stringfrontier_result="[(4:270)]\n"
"fail\n""1\n" Tests;
```

```
 2
 3  General.Node;
 4  Algorithms.PartA_BFS;
 5
 6   org.junit.Assert.assertEquals;
 7
 8  java.io.ByteArrayOutputStream;
 9  java.io.PrintStream;
10
11  org.junit.Test;
12
13    {
14
15           ();
16
17
18         {
19          System.setOut( (outContent));
20
21              (, , , , );
22              (, , , , goal);
23             ;
24
25                   +
26                   +
27                   +
28                   +
29                   +
30                   +
31                   +
32                   +
33                   +
34                   + ;
35
36          PartA_BFS.bfs(start, goal, planetSize);
37          assertEquals(frontier_result, outContent.toString());
```

```
    }


       {
        System.setOut( (outContent));

            (, , , , );
            (, , , , goal);
           ;

                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                +
                + ;

        PartA_BFS.bfs(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }


```

```
        {
         System.setOut( (outContent));

             (, , , , );
             (, , , , goal);
            ;

                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  + ;

         PartA_BFS.bfs(start, goal, planetSize);
         assertEquals(frontier_result, outContent.toString());
    }


        {
         System.setOut( (outContent));

             (, , , , );
             (, , , , goal);
            ;

                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  +
                  + ;

         PartA_BFS.bfs(start, goal, planetSize);
         assertEquals(frontier_result, outContent.toString());
    }


```

```
160         {
161          System.setOut( (outContent));
162 
163              (, , , , );
164              (, , , , goal);
165             ;
166 
167                  +
168                  +
169                  +
170                  +
171                  +
172                  +
173                  + ;
174 
175          PartA_BFS.bfs(start, goal, planetSize);
176          assertEquals(frontier_result, outContent.toString());
177      }
178 
179 
180         {
181          System.setOut( (outContent));
182 
183              (, , , , );
184              (, , , , goal);
185             ;
186 
187                  +
188                  + ;
189 
190          PartA_BFS.bfs(start, goal, planetSize);
191          assertEquals(frontier_result, outContent.toString());
192      }
193 
194 
195         {
196          System.setOut( (outContent));
197 
198              (, , , , );
199              (, , , , goal);
200             ;
201 
202                  +
203                  +
204                  +
205                  +
206                  +
207                  +
208                  +
209                  +
210                  + ;
211 
212          PartA_BFS.bfs(start, goal, planetSize);
213          assertEquals(frontier_result, outContent.toString());
214      }
215 
216 
217         {
218          System.setOut( (outContent));
219 
220              (, , , , );
```

```
            (, , , , goal);
           ;

                +
                + ;

        PartA_BFS.bfs(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }
}
```