src/Tests/PartB_AStarTests.java

```
  1 packageimportstaticimportimportimportimportimportpublicclass
PartB_AStarTestsprivatefinalByteArrayOutputStreamoutContent=new
ByteArrayOutputStream@Testpublicvoidtest BasicPathfinding()newPrintStreamNode
goal=newNode1180null0nullNodestart=newNode190null0intplanetSize=4String
frontier_result="[(1:90)1.414]\n""[(1:135)1.551,(1:45)2.633,(2:90)3.236]\n"
"[(1:180)2.356,(1:45)2.633,(2:90)3.236,(2:135)4.044]\n""(1:90)(1:135)
(1:180)\n""1.571\n""3\n"@Testpublicvoidtest AdvancedPathfinding()newPrintStream
Nodegoal=newNode290null0nullNodestart=newNode70null0intplanetSize=8String
frontier_result="[(7:0)7.280]\n""[(6:0)7.325,(7:45)11.260,(7:315)14.030]\n"
"[(5:0)8.385,(7:45)11.260,(6:45)11.511,(7:315)14.030,(6:315)14.260]\n"
"[(4:0)10.472,(7:45)11.260,(6:45)11.511,(5:45)12.782,(7:315)14.030,
(6:315)14.260,(5:315)15.495]\n""[(7:45)11.260,(6:45)11.511,(5:45)12.782,
(3:0)13.606,(7:315)14.030,(6:315)14.260,(4:45)15.089,(5:315)15.495,
(4:315)17.737]\n""[(6:45)11.511,(5:45)12.782,(3:0)13.606,(7:315)14.030,
(6:315)14.260,(4:45)15.089,(5:315)15.495,(4:315)17.737,(7:90)21.493]\n"
"[(5:45)12.782,(3:0)13.606,(7:315)14.030,(6:315)14.260,(4:45)15.089,
(5:315)15.495,(4:315)17.737,(6:90)21.137,(7:90)21.493]\n""[(3:0)13.606,
(7:315)14.030,(6:315)14.260,(4:45)15.089,(5:315)15.495,(4:315)17.737,
(6:90)21.137,(7:90)21.493,(5:90)21.781]\n""[(7:315)14.030,(6:315)14.260,
(4:45)15.089,(5:315)15.495,(4:315)17.737,(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781]\n""[(6:315)14.260,
(4:45)15.089,(5:315)15.495,(4:315)17.737,(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(7:270)25.493]\n"
"[(4:45)15.089,(5:315)15.495,(4:315)17.737,(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(6:270)25.137,
(7:270)25.493]\n""[(5:315)15.495,(4:315)17.737,(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(4:90)23.425,
(6:270)25.137,(7:270)25.493]\n""[(4:315)17.737,(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(4:90)23.425,
(6:270)25.137,(7:270)25.493,(5:270)25.781]\n""[(2:0)17.828,(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(4:90)23.425,
(6:270)25.137,(7:270)25.493,(5:270)25.781,(4:270)27.425]\n""[(3:45)18.481,
(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,(2:45)23.102,(1:0)23.236,
(4:90)23.425,(6:270)25.137,(2:315)25.266,(7:270)25.493,(5:270)25.781,
(4:270)27.425]\n""[(3:315)20.991,(6:90)21.137,(7:90)21.493,(5:90)21.781,
(2:45)23.102,(1:0)23.236,(4:90)23.425,(6:270)25.137,(2:315)25.266,
(7:270)25.493,(5:270)25.781,(3:90)26.069,(4:270)27.425]\n""[(6:90)21.137,
(7:90)21.493,(5:90)21.781,(2:45)23.102,(1:0)23.236,(4:90)23.425,(6:270)25.137,
(2:315)25.266,(7:270)25.493,(5:270)25.781,(3:90)26.069,(4:270)27.425,
(3:270)30.069]\n""[(7:90)21.493,(5:90)21.781,(2:45)23.102,(1:0)23.236,
(4:90)23.425,(6:270)25.137,(2:315)25.266,(7:270)25.493,(5:270)25.781,
(3:90)26.069,(4:270)27.425,(3:270)30.069,(6:135)37.073]\n""[(5:90)21.781,
(2:45)23.102,(1:0)23.236,(4:90)23.425,(6:270)25.137,(2:315)25.266,
(7:270)25.493,(5:270)25.781,(3:90)26.069,(4:270)27.425,(3:270)30.069,
(6:135)37.073,(7:135)38.749]\n""[(2:45)23.102,(1:0)23.236,(4:90)23.425,
(6:270)25.137,(2:315)25.266,(7:270)25.493,(5:270)25.781,(3:90)26.069,
(4:270)27.425,(3:270)30.069,(5:135)36.417,(6:135)37.073,(7:135)38.749]\n"
"[(1:0)23.236,(4:90)23.425,(6:270)25.137,(2:315)25.266,(7:270)25.493,
(5:270)25.781,(3:90)26.069,(4:270)27.425,(2:90)29.712,(3:270)30.069,
(1:45)30.615,(5:135)36.417,(6:135)37.073,(7:135)38.749]\n""[(4:90)23.425,
(6:270)25.137,(2:315)25.266,(7:270)25.493,(5:270)25.781,(3:90)26.069,
(4:270)27.425,(1:45)29.259,(2:90)29.712,(3:270)30.069,(1:315)30.583,
(1:45)30.615,(5:135)36.417,(6:135)37.073,(7:135)38.749]\n""[(6:270)25.137,
(2:315)25.266,(7:270)25.493,(5:270)25.781,(3:90)26.069,(4:270)27.425,
(1:45)29.259,(2:90)29.712,(3:270)30.069,(1:315)30.583,(1:45)30.615,
(5:135)36.417,(4:135)36.797,(6:135)37.073,(7:135)38.749]\n""[(2:315)25.266,
(7:270)25.493,(5:270)25.781,(3:90)26.069,(4:270)27.425,(1:45)29.259,
(2:90)29.712,(3:270)30.069,(1:315)30.583,(1:45)30.615,(5:135)36.417,
```

```
(4:135)36.797,(6:135)37.073,(7:135)38.749,(6:225)39.822]\n""[(7:270)25.493,
(5:270)25.781,(3:90)26.069,(4:270)27.425,(1:45)29.259,(2:90)29.712,
(3:270)30.069,(1:315)30.583,(1:45)30.615,(2:270)33.712,(5:135)36.417,
(4:135)36.797,(6:135)37.073,(7:135)38.749,(6:225)39.822]\n""[(5:270)25.781,
(3:90)26.069,(4:270)27.425,(1:45)29.259,(2:90)29.712,(3:270)30.069,
(1:315)30.583,(1:45)30.615,(2:270)33.712,(5:135)36.417,(4:135)36.797,
(6:135)37.073,(7:135)38.749,(6:225)39.822,(7:225)41.519]\n""[(3:90)26.069,
(4:270)27.425,(1:45)29.259,(2:90)29.712,(3:270)30.069,(1:315)30.583,
(1:45)30.615,(2:270)33.712,(5:135)36.417,(4:135)36.797,(6:135)37.073,
(7:135)38.749,(5:225)39.130,(6:225)39.822,(7:225)41.519]\n""[(4:270)27.425,
(1:45)29.259,(2:90)29.712,(3:270)30.069,(1:315)30.583,(1:45)30.615,
(2:270)33.712,(5:135)36.417,(4:135)36.797,(6:135)37.073,(3:135)38.262,
(7:135)38.749,(5:225)39.130,(6:225)39.822,(7:225)41.519]\n""[(1:45)29.259,
(2:90)29.712,(3:270)30.069,(1:315)30.583,(1:45)30.615,(2:270)33.712,
(5:135)36.417,(4:135)36.797,(6:135)37.073,(3:135)38.262,(7:135)38.749,
(5:225)39.130,(4:225)39.445,(6:225)39.822,(7:225)41.519]\n""[(2:90)29.712,
(3:270)30.069,(1:315)30.583,(1:45)30.615,(2:270)33.712,(1:90)36.356,
(5:135)36.417,(4:135)36.797,(6:135)37.073,(3:135)38.262,(7:135)38.749,
(5:225)39.130,(4:225)39.445,(6:225)39.822,(7:225)41.519]\n""(7:0)(6:0)(5:0)
(4:0)(3:0)(2:0)(2:45)(2:90)\n""8.142\n""30\n"@Testpublicvoid
testAdvancedPathfinding2()newPrintStreamNodegoal=newNode10null0nullNodestart=
newNode3180null0intplanetSize=4Stringfrontier_result="[(3:180)4.000]\n"
"[(2:180)4.000,(3:135)6.130,(3:225)6.130]\n""[(1:180)5.000,(3:135)6.130,
(3:225)6.130,(2:135)6.369,(2:225)6.369]\n""[(3:135)6.130,(3:225)6.130,
(2:135)6.369,(2:225)6.369,(1:135)7.633,(1:225)7.633]\n""[(3:225)6.130,
(2:135)6.369,(2:225)6.369,(1:135)7.633,(1:225)7.633,(3:90)10.231]\n"
"[(2:135)6.369,(2:225)6.369,(1:135)7.633,(1:225)7.633,(3:90)10.231,
(3:270)10.231]\n""[(2:225)6.369,(1:135)7.633,(1:225)7.633,(2:90)9.948,
(3:90)10.231,(3:270)10.231]\n""[(1:135)7.633,(1:225)7.633,(2:90)9.948,
(2:270)9.948,(3:90)10.231,(3:270)10.231]\n""[(1:225)7.633,(2:90)9.948,
(2:270)9.948,(3:90)10.231,(3:270)10.231,(1:90)10.770]\n""[(2:90)9.948,
(2:270)9.948,(3:90)10.231,(3:270)10.231,(1:90)10.770,(1:270)10.770]\n"
"[(2:270)9.948,(3:90)10.231,(3:270)10.231,(1:90)10.770,(1:270)10.770,
(2:45)14.898]\n""[(3:90)10.231,(3:270)10.231,(1:90)10.770,(1:270)10.770,
(2:45)14.898,(2:315)14.898]\n""[(3:270)10.231,(1:90)10.770,(1:270)10.770,
(2:45)14.898,(2:315)14.898,(3:45)16.537]\n""[(1:90)10.770,(1:270)10.770,
(2:45)14.898,(2:315)14.898,(3:45)16.537,(3:315)16.537]\n""[(1:270)10.770,
(1:45)14.478,(2:45)14.898,(2:315)14.898,(3:45)16.537,(3:315)16.537]\n"
"[(1:45)14.478,(1:315)14.478,(2:45)14.898,(2:315)14.898,(3:45)16.537,
(3:315)16.537]\n""[(1:315)14.478,(2:45)14.898,(2:315)14.898,(3:45)16.537,
(3:315)16.537,(1:0)18.854]\n""[(2:45)14.898,(2:315)14.898,(3:45)16.537,
(3:315)16.537,(1:0)18.854]\n""[(2:315)14.898,(3:45)16.537,(3:315)16.537,
(1:0)18.854,(2:0)21.708]\n""[(3:45)16.537,(3:315)16.537,(1:0)18.854,
(2:0)21.708]\n""[(3:315)16.537,(1:0)18.854,(2:0)21.708,(3:0)25.562]\n"
"[(1:0)18.854,(2:0)21.708,(3:0)25.562]\n""(3:180)(2:180)(1:180)(1:135)(1:90)
(1:45)(1:0)\n""5.142\n""22\n"@Testpublicvoidtest GoalOfZero()newPrintStreamNode
goal=newNode00null0nullNodestart=newNode190null0intplanetSize=2String
frontier_result="[(1:90)1.000]\n""[(1:45)1.785,(1:135)1.785]\n""[(1:135)1.785,
(1:0)3.356]\n""[(1:0)3.356,(1:180)3.356]\n""[(1:180)3.356,(1:315)5.712]\n"
"[(1:225)5.712,(1:315)5.712]\n""[(1:315)5.712,(1:270)8.854]\n"
"[(1:270)8.854]\n""fail\n""8\n"@Testpublicvoidtest GoalOutOfBounds()new
PrintStreamNodegoal=newNode490null0nullNodestart=newNode190null0intplanetSize=
2Stringfrontier_result="[(1:90)3.000]\n""[(1:45)4.153,(1:135)4.153]\n"
"[(1:135)4.153,(1:0)6.479]\n""[(1:0)6.479,(1:180)6.479]\n""[(1:180)6.479,
(1:315)9.472]\n""[(1:225)9.472,(1:315)9.472]\n""[(1:315)9.472,
(1:270)12.854]\n""[(1:270)12.854]\n""fail\n""8\n"@Testpublicvoid
testStartOutOfBounds()newPrintStreamNodegoal=newNode190null0nullNodestart=new
Node490null0intplanetSize=2Stringfrontier_result="[(4:90)3.000]\n""fail\n"
"1\n" Tests;
  2
  3   org.junit.Assert.assertEquals;
```

```
java.io.ByteArrayOutputStream;
java.io.PrintStream;

org.junit.Test;

Algorithms.PartB_AStar;
General.Node;

  {

         ();


     {
      System.setOut( (outContent));

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

               +
               +
               +
               +
               + ;

      PartB_AStar.AStar(start, goal, planetSize);
      assertEquals(frontier_result, outContent.toString());
   }


     {
      System.setOut( (outContent));

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

               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
               +
```

```
                +
                +
                +
                +
                +
                +
                +
                +
                +
                + ;

        PartB_AStar.AStar(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }


       {
        System.setOut( (outContent));

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

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

        PartB_AStar.AStar(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }


       {
        System.setOut( (outContent));

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

                +
```

```
                +
                +
                +
                +
                +
                +
                +
                + ;

        PartB_AStar.AStar(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }


       {
        System.setOut( (outContent));

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

                +
                +
                +
                +
                +
                +
                +
                +
                + ;

        PartB_AStar.AStar(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }


       {
        System.setOut( (outContent));

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

                +
                + ;

        PartB_AStar.AStar(start, goal, planetSize);
        assertEquals(frontier_result, outContent.toString());
    }
}

```