[陈](https://www.coursera.org/" \o "主页)



返回到第 4 周**课程**

[上一个](https://www.coursera.org/learn/algorithms-part1/quiz/rOobS/interview-questions-optional)

[下一个](https://www.coursera.org/learn/algorithms-part1/supplement/2kwpU/lecture-slides)

### Priority Queues

##### [Overview](https://www.coursera.org/learn/algorithms-part1/supplement/vVzFp/overview)

[10 分](https://www.coursera.org/learn/algorithms-part1/supplement/vVzFp/overview)

##### [Lecture Slides](https://www.coursera.org/learn/algorithms-part1/supplement/eHe3d/lecture-slides)

##### [APIs and Elementary Implementations](https://www.coursera.org/learn/algorithms-part1/lecture/A3kA3/apis-and-elementary-implementations)

[12 分](https://www.coursera.org/learn/algorithms-part1/lecture/A3kA3/apis-and-elementary-implementations)

##### [Binary Heaps](https://www.coursera.org/learn/algorithms-part1/lecture/Uzwy6/binary-heaps)

[23 分](https://www.coursera.org/learn/algorithms-part1/lecture/Uzwy6/binary-heaps)

##### [Heapsort](https://www.coursera.org/learn/algorithms-part1/lecture/ZjoSM/heapsort)

[14 分](https://www.coursera.org/learn/algorithms-part1/lecture/ZjoSM/heapsort)

##### [Event-Driven Simulation (optional)](https://www.coursera.org/learn/algorithms-part1/lecture/QVhGs/event-driven-simulation-optional)

[22 分](https://www.coursera.org/learn/algorithms-part1/lecture/QVhGs/event-driven-simulation-optional)

##### ****[练习测验:](https://www.coursera.org/learn/algorithms-part1/quiz/rOobS/interview-questions-optional)****

##### [Interview Questions (optional)](https://www.coursera.org/learn/algorithms-part1/quiz/rOobS/interview-questions-optional)

[3 questions](https://www.coursera.org/learn/algorithms-part1/quiz/rOobS/interview-questions-optional)

##### ****[编程作业:](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle)****

##### [8 Puzzle](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle)

[5小时](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle)

### Elementary Symbol Tables

## 编程作业: 8 Puzzle

您还未提交解答。您必须获得 80/100 分才能通过。

|  |  |
| --- | --- |
| **截止时间** | Pass this assignment by 二月 19, 11:59 晚上 PST |

1. [说明](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle)
2. [**我提交的作业**](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle/submission)
3. [讨论](https://www.coursera.org/learn/algorithms-part1/programming/iqOQi/8-puzzle/discussions)

  Create submission

### Your Submissions

**日期**

**分数**

**通过了吗？**

25 一月 2017 在 9:24 晚上

100/100

是

8 Puzzle

100/100

隐藏 评分反馈

See the Assessment Guide for information on how to interpret this report.

ASSESSMENT SUMMARY

Compilation: PASSED

API: PASSED

Findbugs: FAILED (2 warnings)

Checkstyle: FAILED (16 warnings)

Correctness: 42/42 tests passed

Memory: 11/11 tests passed

Timing: 17/17 tests passed

Aggregate score: 100.00%

[Compilation: 5%, API: 5%, Findbugs: 0%, Checkstyle: 0%, Correctness: 60%, Memory: 10%, Timing: 20%]

ASSESSMENT DETAILS

The following files were submitted:

----------------------------------

6.7K Jan 25 13:24 Board.java

4.4K Jan 25 13:24 Solver.java

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* COMPILING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% javac Board.java

\*-----------------------------------------------------------

% javac Solver.java

\*-----------------------------------------------------------

================================================================

Checking the APIs of your programs.

\*-----------------------------------------------------------

Board:

Solver:

================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* CHECKING STYLE AND COMMON BUG PATTERNS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% findbugs \*.class

\*-----------------------------------------------------------

L B NM\_CLASS\_NAMING\_CONVENTION Nm: The class name 'Solver$boardNode' doesn't start with an upper-case letter. At Solver.java:[lines 23-32]

L B NM\_CLASS\_NAMING\_CONVENTION Nm: The class name 'Solver$boardNodeComparator' doesn't start with an upper-case letter. At Solver.java:[lines 35-37]

Warnings generated: 2

================================================================

% checkstyle \*.java

\*-----------------------------------------------------------

Board.java:4:8: Unused import statement for 'edu.princeton.cs.algs4.StdOut'. [UnusedImports]

Board.java:5:8: Unused import statement for 'edu.princeton.cs.algs4.StdRandom'. [UnusedImports]

Board.java:74:25: Inner assignments should be avoided. [InnerAssignment]

Board.java:107:17: Do not use the 'instanceof' operator in this course. Use 'getClass()' to compare classes. [IllegalToken]

Board.java:176:16: 'for' is not followed by whitespace. [WhitespaceAfter]

Solver.java:14:23: The instance variable 'twin\_iniNode' must start with a lowercase letter and use camelCase. [MemberName]

Solver.java:16:30: The instance variable 'twin\_pQueue' must start with a lowercase letter and use camelCase. [MemberName]

Solver.java:18:31: The instance variable 'twin\_deleted' must start with a lowercase letter and use camelCase. [MemberName]

Solver.java:23:19: The class 'boardNode' must start with an uppercase letter and use CamelCase. [TypeName]

Solver.java:35:19: The class 'boardNodeComparator' must start with an uppercase letter and use CamelCase. [TypeName]

Solver.java:48:44: Inner assignments should be avoided. [InnerAssignment]

Solver.java:49:50: Inner assignments should be avoided. [InnerAssignment]

Solver.java:50:42: Inner assignments should be avoided. [InnerAssignment]

Solver.java:60:19: The local variable 'twin\_del' must start with a lowercase letter and use camelCase. [LocalVariableName]

Solver.java:61:22: Inner assignments should be avoided. [InnerAssignment]

Solver.java:62:31: Inner assignments should be avoided. [InnerAssignment]

Checkstyle ends with 16 errors.

================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TESTING CORRECTNESS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing correctness of Board

\*-----------------------------------------------------------

Running 22 total tests.

Tests 5, 6, 13, and 14 rely upon toString() returning results in prescribed format.

Test 1a: Test hamming() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle07.txt

\* puzzle17.txt

\* puzzle27.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 1b: Test hamming() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

\* 127-by-127

==> passed

Test 2a: Test manhattan() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle07.txt

\* puzzle17.txt

\* puzzle27.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 2b: Test manhattan() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

\* 127-by-127

==> passed

Test 3: Test dimension() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

==> passed

Test 4a: Test toString() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle06.txt

\* puzzle09.txt

\* puzzle23.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 4b: Test toString() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

\* 127-by-127

==> passed

Test 5a: Test neighbors() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle06.txt

\* puzzle09.txt

\* puzzle23.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 5b: Test neighbors() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

\* 127-by-127

==> passed

Test 6a: Test neighbors() of neigbors() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle06.txt

\* puzzle09.txt

\* puzzle23.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 6b: Test neighbors() of neighbors() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

==> passed

Test 7a: Test twin() with file inputs

\* puzzle04.txt

\* puzzle00.txt

\* puzzle06.txt

\* puzzle09.txt

\* puzzle23.txt

\* puzzle2x2-unsolvable1.txt

==> passed

Test 7b: Test twin() with random n-by-n boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 9-by-9

\* 10-by-10

==> passed

Test 8a: Test isGoal() on file inputs

\* puzzle00.txt

\* puzzle04.txt

\* puzzle16.txt

\* puzzle06.txt

\* puzzle09.txt

\* puzzle23.txt

\* puzzle2x2-unsolvable1.txt

\* puzzle3x3-unsolvable1.txt

\* puzzle3x3-00.txt

\* puzzle4x4-00.txt

==> passed

Test 8b: Test isGoal() on n-by-n goal boards

\* 2-by-2

\* 3-by-3

\* 4-by-4

\* 5-by-5

\* 6-by-6

\* 100-by-100

==> passed

Test 9: Check whether two Board objects can be created at the same time

\* random 3-by-3 and 3-by-3 boards

\* random 4-by-4 and 4-by-4 boards

\* random 2-by-2 and 2-by-2 boards

\* random 3-by-3 and 4-by-4 boards

\* random 4-by-4 and 3-by-3 boards

==> passed

Test 10a: Check equals()

\* reflexive

\* symmetric

\* checks that individual entries of array are equal

\* argument is object of type String

\* argument is object of type Object

\* argument is null

\* argument is Board of different dimension

==> passed

Test 10b: Test equals() on m-by-m vs. n-by-n goal boards

\* m = 2, n = 2

\* m = 3, n = 3

\* m = 4, n = 4

\* m = 2, n = 5

\* m = 5, n = 2

==> passed

Test 11: Check that Board is immutable by changing argument array after

construction and making sure Board does not mutate

==> passed

Test 12: Check that Board is immutable by testing whether methods

return the same value, regardless of order in which called

\* puzzle10.txt

\* puzzle20.txt

\* puzzle30.txt

\* 2-by-2

\* 3-by-3

\* 4-by-4

==> passed

Test 13: Call hamming() on a board that is kth-neighbor of a board

\* 0th neighbor of puzzle27.txt

\* 1th neighbor of puzzle27.txt

\* 2th neighbor of puzzle27.txt

\* 13th neighbor of puzzle27.txt

\* 13th neighbor of puzzle00.txt

\* 13th neighbor of puzzle2x2-unsolvable1.txt

==> passed

Test 14: Call manhattan() on a board that is a kth-neighbor of a board

\* 0th neighbor of puzzle27.txt

\* 1th neighbor of puzzle27.txt

\* 2th neighbor of puzzle27.txt

\* 13th neighbor of puzzle27.txt

\* 13th neighbor of puzzle00.txt

\* 13th neighbor of puzzle2x2-unsolvable1.txt

==> passed

Total: 22/22 tests passed!

================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TESTING CORRECTNESS (substituting reference Board)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing correctness of Solver

\*-----------------------------------------------------------

Running 20 total tests.

Test 1: Call moves() with file inputs

\* puzzle00.txt

\* puzzle01.txt

\* puzzle02.txt

\* puzzle03.txt

\* puzzle04.txt

\* puzzle05.txt

\* puzzle06.txt

\* puzzle07.txt

\* puzzle08.txt

\* puzzle09.txt

\* puzzle10.txt

\* puzzle11.txt

\* puzzle12.txt

\* puzzle13.txt

==> passed

Test 2: Call solution() with file inputs

\* puzzle00.txt

\* puzzle01.txt

\* puzzle02.txt

\* puzzle03.txt

\* puzzle04.txt

\* puzzle05.txt

\* puzzle06.txt

\* puzzle07.txt

\* puzzle08.txt

\* puzzle10.txt

\* puzzle15.txt

==> passed

Test 3: Create two Solver objects at the same time

\* puzzle04.txt and puzzle04.txt

\* puzzle00.txt and puzzle04.txt

\* puzzle04.txt and puzzle00.txt

==> passed

Test 4a: Call isSolvable() with file inputs

\* puzzle01.txt

\* puzzle03.txt

\* puzzle04.txt

\* puzzle17.txt

\* puzzle3x3-unsolvable1.txt

\* puzzle3x3-unsolvable2.txt

\* puzzle4x4-unsolvable.txt

==> passed

Test 4b: Call isSolvable() on random n-by-n boards

\* 100 random 2-by-2 boards

==> passed

Test 5: Call moves() on unsolvable puzzles

\* puzzle2x2-unsolvable1.txt

\* puzzle2x2-unsolvable2.txt

\* puzzle3x3-unsolvable1.txt

\* puzzle3x3-unsolvable2.txt

\* puzzle4x4-unsolvable.txt

==> passed

Test 6: Call solution() on unsolvable puzzles

\* puzzle2x2-unsolvable1.txt

\* puzzle2x2-unsolvable2.txt

\* puzzle3x3-unsolvable1.txt

\* puzzle3x3-unsolvable2.txt

\* puzzle4x4-unsolvable.txt

==> passed

Test 7a: Check that Solver is immutable by testing whether methods

return the same value, regardless of order in which called

\* puzzle3x3-00.txt

\* puzzle3x3-01.txt

\* puzzle3x3-05.txt

\* puzzle3x3-10.txt

\* random 2-by-2 solvable boards

==> passed

Test 7b: Check that Solver is immutable by testing whether methods

return the same value, regardless of order in which called

\* puzzle3x3-unsolvable1.txt

\* puzzle3x3-unsolvable2.txt

\* puzzle4x4-unsolvable.txt

\* random 2-by-2 unsolvable boards

==> passed

Test 8: Call moves() with more file inputs

\* puzzle14.txt

\* puzzle15.txt

\* puzzle16.txt

\* puzzle17.txt

\* puzzle18.txt

\* puzzle19.txt

\* puzzle20.txt

\* puzzle21.txt

\* puzzle22.txt

\* puzzle23.txt

\* puzzle24.txt

\* puzzle25.txt

\* puzzle26.txt

\* puzzle27.txt

\* puzzle28.txt

\* puzzle29.txt

\* puzzle30.txt

\* puzzle31.txt

==> passed

Test 9: Check whether equals() method in Board is called

with an argument of the wrong type

\* puzzle00.txt

\* puzzle05.txt

\* puzzle10.txt

\* puzzle15.txt

==> passed

Test 10: Check that constructor throws exception if board is null

==> passed

Test 11: Check for fragile dependence on toString()

\* puzzle00.txt

\* puzzle04.txt

\* puzzle08.txt

\* puzzle12.txt

==> passed

Test 12a: Call moves() with 2-by-2 file inputs

\* puzzle2x2-00.txt

\* puzzle2x2-01.txt

\* puzzle2x2-02.txt

\* puzzle2x2-03.txt

\* puzzle2x2-04.txt

\* puzzle2x2-05.txt

\* puzzle2x2-06.txt

==> passed

Test 12b: Call solution() with 2-by-2 file inputs

\* puzzle2x2-00.txt

\* puzzle2x2-01.txt

\* puzzle2x2-02.txt

\* puzzle2x2-03.txt

\* puzzle2x2-04.txt

\* puzzle2x2-05.txt

\* puzzle2x2-06.txt

==> passed

Test 13a: Call moves() with 3-by-3 file inputs

\* puzzle3x3-00.txt

\* puzzle3x3-01.txt

\* puzzle3x3-02.txt

\* puzzle3x3-03.txt

\* puzzle3x3-04.txt

\* puzzle3x3-05.txt

\* puzzle3x3-06.txt

\* puzzle3x3-07.txt

\* puzzle3x3-08.txt

\* puzzle3x3-09.txt

\* puzzle3x3-10.txt

\* puzzle3x3-11.txt

\* puzzle3x3-12.txt

\* puzzle3x3-13.txt

\* puzzle3x3-14.txt

\* puzzle3x3-15.txt

\* puzzle3x3-16.txt

\* puzzle3x3-17.txt

\* puzzle3x3-18.txt

\* puzzle3x3-19.txt

\* puzzle3x3-20.txt

\* puzzle3x3-21.txt

\* puzzle3x3-22.txt

\* puzzle3x3-23.txt

\* puzzle3x3-24.txt

\* puzzle3x3-25.txt

\* puzzle3x3-26.txt

\* puzzle3x3-27.txt

\* puzzle3x3-28.txt

\* puzzle3x3-29.txt

\* puzzle3x3-30.txt

==> passed

Test 13b: Call solution() with 3-by-3 file inputs

\* puzzle3x3-00.txt

\* puzzle3x3-01.txt

\* puzzle3x3-02.txt

\* puzzle3x3-03.txt

\* puzzle3x3-04.txt

\* puzzle3x3-05.txt

\* puzzle3x3-06.txt

\* puzzle3x3-07.txt

\* puzzle3x3-08.txt

\* puzzle3x3-09.txt

\* puzzle3x3-10.txt

\* puzzle3x3-11.txt

\* puzzle3x3-12.txt

\* puzzle3x3-13.txt

\* puzzle3x3-14.txt

\* puzzle3x3-15.txt

\* puzzle3x3-16.txt

\* puzzle3x3-17.txt

\* puzzle3x3-18.txt

\* puzzle3x3-19.txt

\* puzzle3x3-20.txt

\* puzzle3x3-21.txt

\* puzzle3x3-22.txt

\* puzzle3x3-23.txt

\* puzzle3x3-24.txt

\* puzzle3x3-25.txt

\* puzzle3x3-26.txt

\* puzzle3x3-27.txt

\* puzzle3x3-28.txt

\* puzzle3x3-29.txt

\* puzzle3x3-30.txt

==> passed

Test 14a: Call moves() with 4-by-4 file inputs

\* puzzle4x4-00.txt

\* puzzle4x4-01.txt

\* puzzle4x4-02.txt

\* puzzle4x4-03.txt

\* puzzle4x4-04.txt

\* puzzle4x4-05.txt

\* puzzle4x4-06.txt

\* puzzle4x4-07.txt

\* puzzle4x4-08.txt

\* puzzle4x4-09.txt

\* puzzle4x4-10.txt

\* puzzle4x4-11.txt

\* puzzle4x4-12.txt

\* puzzle4x4-13.txt

\* puzzle4x4-14.txt

\* puzzle4x4-15.txt

\* puzzle4x4-16.txt

\* puzzle4x4-17.txt

\* puzzle4x4-18.txt

\* puzzle4x4-19.txt

\* puzzle4x4-20.txt

\* puzzle4x4-21.txt

\* puzzle4x4-22.txt

\* puzzle4x4-23.txt

\* puzzle4x4-24.txt

\* puzzle4x4-25.txt

\* puzzle4x4-26.txt

\* puzzle4x4-27.txt

\* puzzle4x4-28.txt

\* puzzle4x4-29.txt

\* puzzle4x4-30.txt

==> passed

Test 14b: Call solution() with 4-by-4 file inputs

\* puzzle4x4-00.txt

\* puzzle4x4-01.txt

\* puzzle4x4-02.txt

\* puzzle4x4-03.txt

\* puzzle4x4-04.txt

\* puzzle4x4-05.txt

\* puzzle4x4-06.txt

\* puzzle4x4-07.txt

\* puzzle4x4-08.txt

\* puzzle4x4-09.txt

\* puzzle4x4-10.txt

\* puzzle4x4-11.txt

\* puzzle4x4-12.txt

\* puzzle4x4-13.txt

\* puzzle4x4-14.txt

\* puzzle4x4-15.txt

\* puzzle4x4-16.txt

\* puzzle4x4-17.txt

\* puzzle4x4-18.txt

\* puzzle4x4-19.txt

\* puzzle4x4-20.txt

\* puzzle4x4-21.txt

\* puzzle4x4-22.txt

\* puzzle4x4-23.txt

\* puzzle4x4-24.txt

\* puzzle4x4-25.txt

\* puzzle4x4-26.txt

\* puzzle4x4-27.txt

\* puzzle4x4-28.txt

\* puzzle4x4-29.txt

\* puzzle4x4-30.txt

==> passed

Test 15: Call moves() with random solvable n-by-n boards

\* 100 random 2-by-2 boards

\* 200 random 3-by-3 boards that are <= 20 moves from goal

\* 200 random 4-by-4 boards that are <= 20 moves from goal

\* 200 random 5-by-5 boards that are <= 20 moves from goal

==> passed

Total: 20/20 tests passed!

================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* MEMORY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Computing memory of Board

\*-----------------------------------------------------------

Running 8 total tests.

Memory usage of an n-by-n board

n student (bytes) reference (bytes)

----------------------------------------------------------

=> passed 4 248 240

=> passed 8 568 560

=> passed 12 1016 1008

=> passed 16 1592 1584

=> passed 20 2296 2288

=> passed 36 6392 6384

=> passed 72 23096 23088

=> passed 120 61496 61488

==> 8/8 tests passed

Total: 8/8 tests passed!

Student memory = 4.00 n^2 + 32.00 n + 56.00 (R^2 = 1.000)

Reference memory = 4.00 n^2 + 32.00 n + 48.00 (R^2 = 1.000)

================================================================

Computing memory of Solver

\*-----------------------------------------------------------

Running 3 total tests.

Test 1: memory with puzzle20.txt (must be <= 2.0x reference solution)

- memory of student Solver = 4824 bytes

- memory of reference Solver = 4896 bytes

- student / reference = 0.99

==> passed

Test 2: memory with puzzle25.txt (must be <= 2.0x reference solution)

- memory of student Solver = 5912 bytes

- memory of reference Solver = 6056 bytes

- student / reference = 0.98

==> passed

Test 3: memory with puzzle30.txt (must be <= 2.0x reference solution)

- memory of student Solver = 6912 bytes

- memory of reference Solver = 7216 bytes

- student / reference = 0.96

==> passed

Total: 3/3 tests passed!

================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TIMING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Timing Solver

\*-----------------------------------------------------------

Running 17 total tests.

Timing tests use your implementation of Board.java and Solver.java.

Maximum time allowed per puzzle is 10 seconds.

filename N seconds insert() delMin() max PQ size

---------------------------------------------------------------------------------------------

=> passed puzzle20.txt 3 0.02 1510 897 614

=> passed puzzle21.txt 3 0.01 2757 1615 1143

=> passed puzzle22.txt 3 0.01 3083 1833 1251

=> passed puzzle23.txt 3 0.02 5395 3189 2207

=> passed puzzle24.txt 3 0.02 6499 3941 2559

=> passed puzzle25.txt 3 0.03 11096 6579 4518

=> passed puzzle26.txt 3 0.01 5316 3189 2128

=> passed puzzle27.txt 3 0.02 10008 6043 3966

=> passed puzzle28.txt 3 0.04 24032 14375 9658

=> passed puzzle29.txt 3 0.04 24315 14817 9499

=> passed puzzle30.txt 3 0.06 41630 25163 16468

=> passed puzzle31.txt 3 0.08 53698 32505 21194

=> passed puzzle34.txt 4 0.25 254199 121659 132541

=> passed puzzle37.txt 4 0.15 159197 76215 82983

=> passed puzzle39.txt 4 0.14 151455 74157 77299

=> passed puzzle41.txt 5 0.08 76862 32947 43916

=> passed puzzle44.txt 5 0.45 308964 138177 170788

==> 17/17 tests passed

Total: 17/17 tests passed!

================================================================

25 一月 2017 在 9:13 晚上

95/100

是

25 一月 2017 在 9:08 晚上

95/100

是

25 一月 2017 在 8:42 晚上

82/100

是

25 一月 2017 在 8:26 晚上

77/100

否

25 一月 2017 在 6:45 晚上

75/100

否

25 一月 2017 在 6:33 晚上

73/100

否

8 Puzzle

73/100

显示 评分反馈

25 一月 2017 在 6:23 晚上

70/100

否

25 一月 2017 在 1:33 下午

50/100

否