

Assignment 0 - Setup & Toy Problems

Artificial Intelligence

WS 2018

Due: 2018-10-21, 23:59

Andreas Arzt, Verena Haunschmid, Rainer Kelz

General Information

This assignment is supposed to

1. ensure you are properly prepared for this course
2. familiarise you with the submission procedure for the exercise (Moodle course page: <https://moodle.jku.at/jku2015/course/view.php?id=4770>)

1 Handbook and Programming Environment Setup

Go to the **Moodle** page associated with this course, and **download** the **Rainbows and Unicorns handbook**. Go **read** the **handbook**. Really, do yourself a favor and do it.

If you did manage to

1. download the assignment from Moodle
2. run all the unit tests

you will have noticed that all tests for `assignment0` are failing. Fixing this problem is part of this assignment.

2 Report for the theoretical part

For handing in your solutions to the theoretical part, we would like you to create a PDF file in the directory named 'reports'. Ideally you would use Latex to produce your reports, but exporting a PDF from any other program is fine too.

Where to put your reports:

```
<your-exercise-directory>
├── reports
│   └── report.pdf (you need to put your theory answers into this file)
├── assets
├── libs
└── src
```

IMPORTANT:

- Your report needs to **contain** your **names** and **immatriculation numbers**!
- **No photos** of handwritten reports!

3 (Programming) Locating the error

Locate the class `at.jku.cp.ai.search.algorithms.RS` which should be here:

```
<your-exercise-directory>
├── src
│   ├── at
│   │   ├── jku
│   │   │   ├── cp
│   │   │   │   ├── ai
│   │   │   │   │   ├── search
│   │   │   │   │   │   ├── algorithms
│   │   │   │   │   │   └── RS.java
```

Locate the lines with `TODO` in the comments. Follow the suggestions in the `TODO` comments. Run the unit tests again, they should be all passing now. If you happen to have a particularly slow computer, you may exceed the timeout that is set for each test. Should this be the case, you may need to increase the time limit - **however**:

The time limit is there for a reason, namely to prevent our test harness to get stuck in an infinite loop you may have inadvertently produced.

All the problem instances you are provided with can be solved in a very reasonable amount of time, with a very reasonable amount of memory with our reference implementations on a (rather old) 2GHz Intel Core Duo T7200. If your tests are failing because of timeouts, and you have a faster processor than that, you should be highly suspicious that there is a bug hiding somewhere. You should be equally suspicious, if your OS decides to use swap space!

(1 point)

4 (Theory) - Datastructures and Runtime Complexity

Let's recap a few things from the first few semesters. These things are actually important when implementing fast search algorithms later in the lecture.

Operation	best	average	worst
insert at the end of a single linked list	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$
...	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$
...	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$	$\mathcal{O}(\dots)$

Table 4.1: Use a table to report your results.

1. What are the best/average/worst case runtime complexities of
 - the `insert` operation at the end of a single linked list?
 - the `insert` operation at the end of a list backed by an array?
 - the `insert` operation of a hash table (hash set)?
 - the `contains` operation of a list?
 - the `contains` operation of a hash table (hash set)?

Use a table like Table 4.1 to report your results.

2. What would be a suitable Java datatype to implement a FIFO-structure?

(6 points)