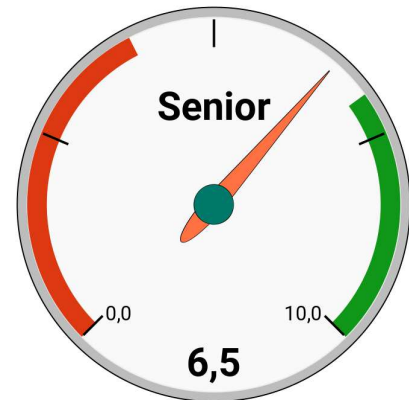


# Java NoSQL

## jnos-1676377863

Mohit

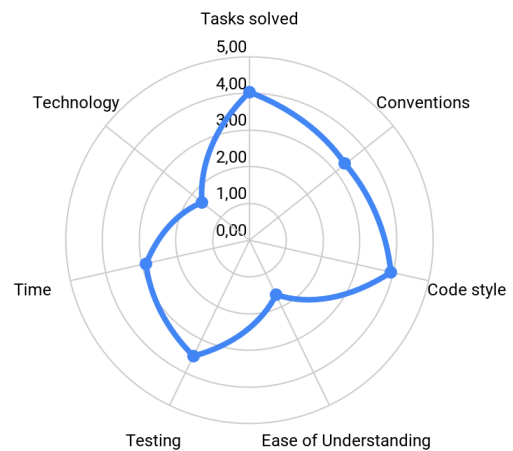


Presented to candidate 09.02.2023 / Review 18.02.2023

**The candidate provided a complete and tested optimal solution with minor issues.**

### EXECUTIVE SUMMARY

- The candidate implemented optimal solutions to all tasks.
- The candidate wrote a good amount of tests.
- The git usage was still ok.
- There were some technological issues, such as no use of the final modifier.
- There was no documentation or whatsoever to guide the user.
- The candidate needed around 10h to solve the tasks



## DETAILS

### Good

- All algorithms were optimal. For the first task the candidate used neighbor-flooding to improve the search speed.
- For the second task, the candidate used BFS to solve the problem.
- The application was built using gradle/maven without problems.
- The candidate used Java 14+ for their solution.
- An interactive CLI was provided to run the tasks.

### Neutral

- The storage to the database used the bytes from the files. Parsing was done after reading the file from the database.
- The tests were collected in just two test methods. This is not optimal and makes debugging harder.
- The branching could be improved. Some branches were stacked upon each other.

### Bad

- The solution methods modified the input. Thus, they do not analyze read-only.
- The candidate did not take advantage of the new switch statements and the “final” modifier was not used at all.
- There were a lot of unused declarations (including some methods)
- The error handling was pretty basic.
- There was no hint or documentation explaining what the app expects. Thus, the mongo-db had to be started by the Reviewer. A Dockerfile or a README would have been useful.
- The code lacked abstraction at some point. There were some duplicated lines for task one and two.