

Index Team Hiring Exercise

Hello and welcome to the Index Team Hiring Exercise. This document introduces you to the context of the exercise, we hope you will find it fun to do.

The exercise is a solve-at-home coding exercise and will be further discussed in a potential follow-up interview.

The exercise task

Your task is to implement a skip list from scratch.

Guidelines

Please implement your solution in Java only. The project is set up to use Java 21, but you may choose any later Java version.

You may not use any external libraries, only what is available in the Java standard library.

You may not use any LLM/AI tools to solve the exercise, nor during the follow-up interview.

Any code submitted as part of the exercise must be written by you as part of the exercise.

If you have any questions about the exercise, please do not hesitate to reach out for clarification. We do not want to judge you on your ability to read our minds, so if anything is unclear, please ask.

Details

If you want to familiarize yourself with skip lists, the Wikipedia article is a good start [\[https://en.wikipedia.org/wiki/Skip_list\]](https://en.wikipedia.org/wiki/Skip_list).

You will receive access to a git repository, hosted on GitHub. You will need to have a GitHub account with [2-factor authentication](#) enabled in order to be able to access this repository. You need to be familiar with git basics to submit the solution.

This repository contains a maven project with an interface for a skip list and a few example tests.

Your first task is to provide an implementation for a skip list for the interface.

You do not have to use the provided maven project, you can build and run the code any way you want.

You can use any non-LLM-enabled editor you like. If your editor comes with settings or plugins for LLM, please disable those before doing the exercise and before the follow-up interview.

Your code will not be used by us and we will delete the repository some time after the interview process has concluded.

Please do not spend masses of your time on this. Provide a simple and straightforward solution over a complicated and thorough one. Don't aim for production-readiness, but try to follow good engineering practices as you see fit.

Building and Running the code

The project is using maven and we provide the maven-wrapper tool in the repository, which will download and use a suitable maven installation, so that you don't have to have maven itself installed.

To build the project, run the following command

```
Shell
# on Unix likes
./mvnw compile
# on Windows
.\mvnw.cmd compile
```

To run the provided test, run the following command

```
Shell
# on Unix likes
./mvnw test
# on Windows
.\mvnw.cmd test
```

Analysis

After the implementation, we would like for you to reflect on it.

Your second task is to provide a written analysis of your implementation.

Please answer these questions:

- Did you make any assumptions that helped you to simplify the code?
- What are the upsides and downsides of your implementation?
- If you needed to scale your solution (say, to billions of entries, or production-readiness), what decisions would you make differently in your approach?
- Do you have any feedback on the exercise in general?

Write the analysis in some text format (plain text, Markdown, AsciiDoc, etc.) and add the file to the repository.

Submission

In order to submit your solution, push your solution and analysis to the repository that was provided to you, then contact your recruiter and let them know that you finished the exercise. You can use any kind of git workflow and commit style that you prefer, but please don't change anything after you have submitted your solution.

You may not push this exercise or your solution to a public repository.

The follow-up interview

Depending on our evaluation of your submission, we will invite you to a follow-up interview.

This interview will be in part structured as a live coding session.

We will try to emulate a normal working day, not an exam, so questions and back-and-forth is encouraged.

We will provide some additional tasks during the interview and want you to continue to work on your solution.

Please make sure that you have your editor and solution set up and available before the interview.

You will need to share your screen during the call. By default we will use Google Meet, please make sure that using Meet and sharing your screen in it will work on your web browser and OS.

Good luck, and have fun!