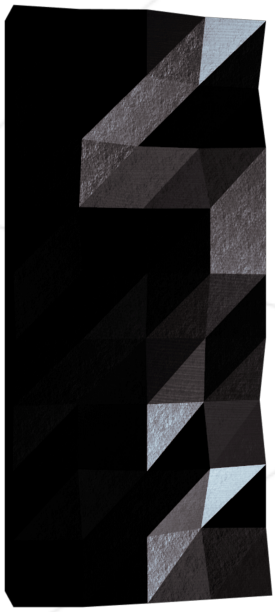

Starfleet

Introduction

Staff 42 pedago@42.fr

Summary: This document is an introduction to the Starfleet Piscine.

Contents

Chapter I

Introduction

Welcome to the Starfleet Piscine !

This piscine is an `introduction to technical interviews`.

It will cover the `basics` required for an `interview` and give you a `good` starting point to futher prepare for technical interviews.

Chapter II

What is a technical interview ?

At most tech companies, `algorithm` and `coding` problems form the largest component of the interview process.

An interview can be on site with a white board or on the phone with a cloud based text editor. In general it can last around 45 minutes, during which you will go through one or several problems.

The interviewer, most likely a software engineer, will `evaluate` your `problem-solving` skills and your `knowledge` of `basic` data structures and algorithms.

The objective of the interview is not to test your knowledge. However, a basic understanding of data structures and algorithms is required to get through some of the problems. That's what we will learn in this piscine!

You can take a look at this video for see how a technical interviews is working (however we advise you to wait the end of the first week to watch it).

Chapter III

Why this piscine ?

III.1 Objective

The objective of this piscine is to solidify your understanding of computer science fundamentals and learn how to apply those fundamentals during an interview.

This piscine is also designed to improve your `communication` skills. You will have to explain your thought process while solving a problem, you might have never seen before.

III.2 New concept

For this piscine, we introduce a new feature : `peer interviews`. During the weekends, you will go through several interviews both as `interviewee` and as an `interviewer`. It is the perfect opportunity for you to walk through a problem using what you have learned during the week, and also put yourself into the interviewer position which is the best way to understand what the interviewer is looking for.

III.3 New Notions

This piscine is a general introduction to algorithms, it will cover a wide variety of notions concerning data structures, algorithms and other fundamentals.

we hope it will make you want to go futher on the `road of optimization` !

Chapter IV

How will it go ?

During the 2 weeks of the Starfleet Piscine, you will go through a set of typical piscine days, exams and interviews.

IV.1 During the week

As usual, piscine days starts at 8:42am and ends the next day at 11:42pm. They are subject to peer correction.

Exams last 2 hours, monday and friday at `6pm`, using the famous `examshell` with brand new exercises.

IV.2 During the week-end (rush)

For the interviews, you will receive an email at the beginning of the day giving you :

- The logins of the 2 students who will interview you.
- The logins of the students you're going to interview. For each student, a pdf will be attached containing the interview question and solutions. As the `interviewer`, it is your job to prepare for the interview (i.e., clearly understand the question and solutions in order to guide the interviewee towards the best solution).

Each interview is `45 minutes` long, guidelines will be provided in the `pdf`. It is up to you to schedule all of the interviews with your `42-fellows`, as long as they all occur during the week-end (before sunday 11:42 pm). At the end of each interview, the interviewer will give a grade on the intranet using the usual grading system.

Chapter V

Where to start ?

Before beginning the D00, take time to learn about the `C standard library` and `Big O notation`.

V.1 C standard library

First, if you don't know the c standard library very well, take time to learn about the functions you can now use!

Basically, a lot of functions that you have implemented in your libft that are in the standard library, like `strcat` or `strdup`.

V.2 Big O notation

Before you start with D00, there is a very important concept you need to focus on : `big O` notation for `time and space complexity`.

`Big O time` is the `language` and metric used to describe the `efficiency of an algorithm`. Not understanding it thoroughly can really `hurt you in developing an algorithm`. Not only might you be judged harshly for not really understanding big O, but you will also struggle to judge when your algorithm is getting faster or slower.

You need to master this concept and be able to evaluate the time and space complexity of any algorithm. Here are some references on common big O notations and the idea behind it :

- Big O explained in video.
- Big O explained in cool stackoverflow post.

Don't worry if you don't fully understand it, `Big O` concept is an `hard topic` that requires a lot of pratice (and you will get it with this piscine!).

V.3 What else ?

Otherwise, we hope you will `enjoy the piscine` !

The Starfleet crew.