

# Tech hiring is hard, and what we do about it

*tl;dr: we worked hard to design an interview process that is fair to us and to candidates, and avoid common annoying pitfall.*

Tech job interview processes are broken and having been on both sides of the process we know what a pain they are.

For candidates:

- Companies lightly asking for hours of investment before disclosing any critical information about the position (e.g.: team, compensation, ...)
- Whiteboard interviews, that put candidates on the spot and are generally only good at assessing one's memory from CS101, and how one react to a particular kind of stress that they'll never encounter again on the job
- Waiting weeks between each step of the process, "*ain't nobody got time for that*".

For recruiters:

- Most applications are shotgun applications and completely irrelevant
- It's really hard to assess someone's skill without actually looking at them doing it
- Resumes rarely indicate the thing we care for

We work hard to do better, and we refined our application process that we believe alleviate some of these problems. It's a fine balance, as we don't want to discard worthy candidates, while making sure we don't waste too much time with irrelevant applications.

## Step 1 - Treasure hunt

Resumes are bad, we generally don't read too much into them (quite literally). Moreover, we found that what people put forward on their resume rarely correlates with what we are looking for in a candidate. Typically, when it comes to hiring of tech people, we look for very strong individual contributors, whereas candidate, especially more senior ones, feel they need to put their leadership experience forward (which, spoiler, we really don't care much about).

We thus designed a treasure hunt: it is a blind (for us) process to screen candidates. Anyone who makes it through gets to talk with us regardless of their gender, race, age, or skills at writing a catchy resume.

We designed the hunt with a few principles in mind:

- it require little time investment and can be achieved in a couple of minutes for the right candidates. *If you stay stuck more than 10 minutes on one step, you're probably not the kind of profile we're looking for.*
- it reflects our company culture: we like nerdy stuff and clever tricks.
- it can be done from anywhere, anytime, at the candidates leisure. We don't keep track of who's going through it.

## Step 2 - Ask Us Anything

The outcome of the treasure hunt is a document with instructions describing the rest of the process. At this stage, the candidate is invited to contact us to ask us questions about the job, the company, or anything else that is more or less appropriate. This is the time for candidates to see if they like us and deem the rest of the process worthy of pursuing.

## Step 3 - Programming Assessment

Candidates are asked to prove they can actually code. If they have some code that they think is good enough, we're very happy to look at it and discuss it. If not, we've designed a short assessment, that shouldn't take more than 2 hours. It's homework and we don't track how long it takes, but we do not want to take advantage of candidates time. We don't expect the perfect solution, but whatever the outcome of those 2 hours is, which we will use it as a starting point for the discussion. *If you can't solve 80% of this fairly simple problem in 2 hours, you may not be the right candidate.*

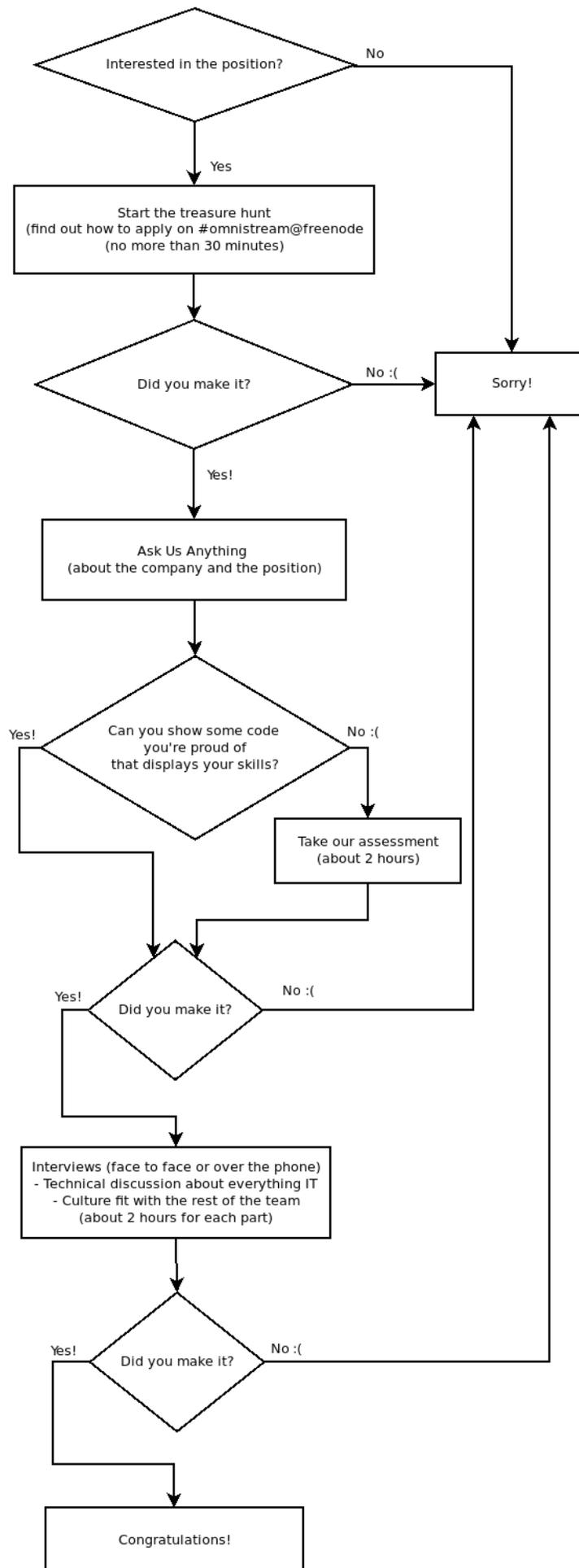
## Step 4 - Final interview

If everything goes fine, candidates are invited in our office (or over video call for remote positions) for a face to face interview. This step has two parts:

1. a technical discussion: we talk about everything CS / IT. Not everyone is expected to know everything, but candidates should be strong in some domains. The outcome of the previous assessment will be discussed (expect questions like: assume you have another 10 hours, how would you improve your solution?). The main point of this discussion is to assess one's broad knowledge and interest in technical domains, we do not expect candidates to prepare anything (e.g.: no need to study *Cracking the code interview* beforehand). Example questions: what's your favorite programming language? Why? Have you heard about \$TECHNOLOGY, what do you think? Did you use it?
2. a "culture fit" interview: The candidate will be expected to talk about something of their choosing for about 10 minutes, with no support. It's really about assessing that the candidate has some basic communication skills, more importantly, that the candidate has passion. As an example, one of us verbalised their passion and knowledge of orange juice, which made for a very interesting discussion. This is also the time to meet the rest of the team, ask more questions and discuss anything that wasn't discussed before to get a good feel for each other.

## That's it

We've made a flowchart to sum it up, because flowcharts are cool. It's a "fail fast" process: if a candidate is not a good fit, it's better for both sides to know about it as early as possible. Moreover, candidates are given enough information early on, before deciding if they want to invest time in the heavier part of the process. We try our best to fit the whole process in less than 2 weeks.



## Where do I start?

The treasure hunt starts on [#omnistream@freenode](#). Good luck!