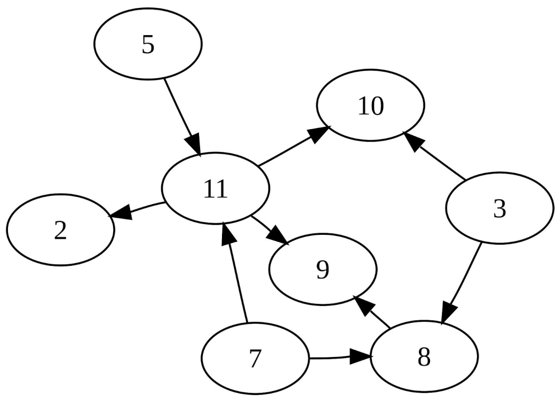
Senior Python Developer Task

Description

Below is the detailed description of the task and that you should think about and implement before the interview.

The goal is to set up a web service that provides an API for clients to run algorithms and perform updates on the graphs. The graphs it will deal with are normal, directed graphs with a set of nodes and edges. There can be cycles in the graph. Each node and edge should have a primary ID and additional metadata like a name and other attributes. The web service needs functions to persist and query the basic graph structure (CRUD methods).

I want you to think about how the graph data structure and the method signatures of the API would look like to do the necessary operations and queries. These methods and data structure should be implemented as a service and provided to clients for consumption. If you run in to time trouble, it is OK to keep data in memory for the implementation and don't do a full data store implementation, but there should be a concept and the data structure ready how you will persist it.



A key requirement is that graphs can be very large, containing hundreds of millions of nodes and edges. Think for example of the Facebook or Twitter social graph. There will be questions in the interview how this can be scaled on a technology level and what an implementation would like like if certain aspects of the graph change. Also the data structure, interfaces and architecture should reflect this and take things like highly concurrent updates, conflicts, sub graph updates and querying, high read or write demands into account that you would expect from such an API. If not everything can be implemented there should be concepts and answers in place in your solution to take care of these challenges.

The second part of the task is to design a client that consumes the graph API to solve a graph problem. The problem is to find the shortest path between two nodes of the graph. Again, important are data structure and this time also how the algorithms works and performs in combination with the API you designed, specifically under the sizing criteria above. Consider if you have to provide more than basic CRUD operations and return data for your type of algorithm in a smarter way to reduce overhead of too many REST calls. Or maybe there are better ways to solve this problem with non naive algorithms or data models when storing the data.

Constraints

You can use any technology of choice for the implementation, just Python as a language is set, but you can choose any database, libraries or other tools around that you feel will help you to get the job done. You can write the client in another language than the server if it is easier and makes sense.

If anything in the description is not specified you can either ask for clarification or are welcome to make an assumption. As long as you can argue for your design choice and assumptions that is fine.

Other Notes

Besides the actual solution, it is also important for the interview that you keep notes for a presentation on how you approached the problem and what problems you encountered. During the interview we would like to get a walkthrough on how you solved the problem, a presentation with some notes and pictures on your thought process and architecture / data model are very helpful there. We will also iterate a bit on your solution and challenge certain assumptions to see how your solution holds up to changes and what you would need to change to make it work under the new assumptions.

Good luck and have fun with it. Get back to me in case anything is unclear or you have questions.