Technical Assessment Case Studies

The purpose of the Case Study is not only to gauge your technical ability, but to see how you think. We will talk through your results here and your logic as part of your on-site interview. The Case Study will not make or break your candidacy for the role, but serve as a single data-point among your other excellent qualifications.

Please complete one of following case studies:

For the case study you choose please meet the following requirements:

* Complete the exercise in the technical stack of your choice.
  + When appropriate use a data store of your choice.
  + Use any external frameworks you desire
  + Be ready to discuss your recommendations to make your solution suitable for use in a production environment
* Provide evidence of the result to the interviewers *(choose one)*
  + Unit test results or other documented output
  + Hosted instance of the implementation
  + Runnable instance of the implementation on your computer
* The end result should be a functional implementation of the problem preferably with associated tests
  + Provide the working code either in a publicly accessible hosted repository or a zip file containing all necessary build steps and dependencies
  + Rename .js files to .js.txt if emailing code
  + Provide a README.md file with instructions for testing, running and interacting with your application and any details you feel are relevant to share
* Please bring either a laptop or a hard copy of the code to help facilitate review at the interview.

1. Document Search

The goal of this exercise is to create a working program to search a set of documents for the given search term or phrase (single token), and return results in order of relevance.

Relevancy is defined as number of times the exact term or phrase appears in the document.

Create three methods for searching the documents:

* Simple string matching
* Text search using regular expressions
* Preprocess the content and then search the index

Prompt the user to enter a search term and search method, execute the search, and return results. For instance:

Enter the search term: <user enters search term>

Search Method: 1) String Match 2) Regular Expression 3) Indexed

Search results:

File2.txt – X matches

File1.txt - X matches

File3.txt – X matches

Elapsed time: 40 ms

Three files have been provided for you to read and use as sample search content.

Run a performance test that does 2M searches with random search terms, and measures execution time. Which approach is fastest? Why?

Provide some thoughts on what you would do on the software or hardware side to make this program scale to handle massive content and/or very large request volume (5000 requests/second or more).