

Code Test

This test has been designed for platform developers or engineers. It is designed to require between two and four hours to complete, but candidates should be allowed up to 48 hours to complete the test. Note: Do not provide the grading scheme to candidates.

1. In your favourite programming language (e.g Java, C#, JavaScript), write code that will find a duplicate value in an array. Do the same for dictionaries.
 - a. Grading: total of 5 points. 1 point for a valid array answer. 1 point for a valid dictionary answer. 1 point for tests (max 2 points). 1 point for style and consistency.
2. In your favourite programming language (e.g Java, C#, JavaScript, Bash), write code that will sort the contents of a file. Assume there is one 32-bit number per line of the file. The file contains 50 billion numbers. The computer that is running this has 8 GB of memory and 256 GB of disk space.
 - a. Grading: 1 point for at least $O(N^2)$ sorting algorithm (e.g. insertion sort, selection sort). 1 point for at least $O(N\log(N))$ sorting algorithm (e.g. merge sort, quicksort, bucket sort). 1 point for storing intermediary sorts on disk (not pulling everything into memory at once). 1 point for tests. 1 point for style and consistency
3. Numerous components on a network, such as DNS servers, switches, load balancers, and others can generate errors anywhere in the life of a given request. The usual technique for dealing with these error responses in a networked environment is to implement retries in the client application. This technique increases the reliability of the application and reduces operational costs for the developer. In your favourite programming language (e.g JavaScript), write code that implements automatic retry logic and simulates exponential backoff and jitter. For the purposes of this code test, assume that the code you are writing will retry on server or throttling errors but will halt on client errors.
 - a. Grading: total of 5 points. 1 point for a simple retry. 1 point for implementing exponential backoff. 1 point for implementing jitter. 1 point for tests. 1 point for style and consistency.
4. Using Java, build a service and a consuming application. The service and the application are packaged as two separate JARs. Your application must consume service JAR and pass JSON string to the service. Service must take that JSON string, convert it to a first-class object, and return the object back to the application.
 - a. Grading: total of 5 points. 1 point for using dependency injection framework, such as Spring or Guice. 1 point for using serialization library such as Gson. 1 point for using Gradle or Maven. 1 point for unit tests. 1 point for style and consistency.
5. A common approach for development is to develop an application programming interface (API) specification for software endpoints. Using software like swagger, create an example API specification that uses at least 4 different HTTP methods for at least 3 different functional endpoints (e.g. health checks, data retrieval, etc).
 - a. Grading: 2 points for a valid openapi specification. 2 points for documentation and details for parameters and response codes. 1 point for metadata.

Total score is ___/25.