SWE Interview Homework Questions

Programming - please pick any 2 of the 3 questions below. Add tests and documentation as you see fit.

Util - 'compare versions'

Implement a utility (in Go, Python, or Java). Feel free to use a framework, if desired.

Compare two version numbers version1 and version2.

If version1 > version2 return 1
If version1 < version2 return -1
otherwise return 0

You may assume that the version strings are non-empty and contain only digits and the 'dot' character. The 'dot' character does not represent a decimal point and is used to separate number sequences. For instance '2.5' is not "two and a half" or "half way to version three", it is the fifth second-level revision of the second first-level revision.

Here is an example of version numbers ordering: 0.1 < 1.1 < 1.2 < 1.2.9.9.9.9 < 1.3 < 1.3.4 < 1.10

Application - 'math api'

Implement a web service (in Go, Python, or Java). Feel free to use a framework, if desired.

/min - given list of numbers and a quantifier (how many) provides min number(s)
/max - given list of numbers and a quantifier (how many) provides max number(s)
/avg - given list of numbers calculates their average
/median - given list of numbers calculates their median
/percentile - given list of numbers and quantifier 'q', compute the qth percentile of the list elements

No need to be concerned with resources, we're assuming there's plenty enough of memory, etc. The percentile should just return one number. Use the nearest-rank method. https://en.wikipedia.org/wiki/Percentile#The_nearest-rank_method

Application - 'tracking web server'

Implement a small small webserver (in Go, Python, or Java).   Feel free to use a framework, if desired.

/ping - returns response code 200 and string OK when file /tmp/ok is present, if file is not present returns 503 service unavailable
/img - returns a 1x1 gif image, and log the request.

Server needs to scale to many concurrent users and be efficient. Propose improvements you'd like to work on and - time permits - implement.