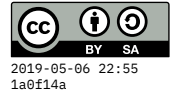


Final exam



This is a **take-home exam**. You may use whatever materials you like, including the source code that we have written and any notes/links provided. If you use quotes or ideas from any source other than your own mind, you **must cite the source** and use quotation marks appropriately.

You may **not** do this exam in groups. Your responses must be entirely your own. If I notice suspicious similarity between the answers of two or more students, or if your answers are copied from another source, your score will suffer.

All of your answers should be **concrete** and pertain **specifically** to the project we implemented this semester. I expect about a page of explanation per question. Submit your answers by email to `christopher.league@liu.edu`.

1. **Feature request:** Most static web servers treat file names as case-sensitive, just as the Linux file system does. However, some servers have configurable options that allow **case-insensitive** path matches – for example `mod_speling` for Apache¹.

If a site is being served with case-insensitivity enabled, it means that a link `` can direct us to a file named `ABOUT.html`.

Is our `webgc` tool case-sensitive or case-insensitive by default? Does that depend on whether it's running on Linux (which has a case-sensitive file system), or on Mac/Windows (which are usually case-insensitive)?

Describe **in detail** what changes would be needed to support a command-line option `-i`, `--ignore-case` that would enable `webgc` to garbage-collect assets correctly when the site is coded with the assumption of case-insensitivity.

2. **Testing:** our `webgc` library uses manual unit tests for several of its modules. Read about property-based testing², and in particular the Python hypothesis library³. Propose a way to apply property-based testing to some aspect of our `webgc` code. Which module would you target first? What are some examples of specific properties that could be verified? (You can write your examples in English; you don't need to write Python code directly to answer this.) How would property-based testing increase your confidence in the code, compared to example-based testing?



¹http://httpd.apache.org/docs/2.2/mod/mod_speling.html



²dev.to/jdsteinhauser/intro-to-property-based-testing-2c18



³hypothesis.readthedocs.io/en/latest/