CMSC162 Intro to Algorithmic Design II Blaheta

Homework 2

Due: 25 September 2023

Problem 2.1

Consider the task of writing functions

```
bool anyEmpty (shared_ptr<Link<string>> chain)
bool allEmpty (shared_ptr<Link<string>> chain)
```

that determine whether any of the strings in a given link chain are empty strings (i.e. `""`), and whether *all* of the strings in a given link chain are empty strings. Both should work correctly on an empty chain (i.e. where chain is a null pointer). (Ask yourself: what *should* they return on an empty chain?)

For the moment I'm more interested in the algorithm, so you're better off writing it out by hand (and I won't fuss over things like semicolons). You should still test your work by tracing it on a concrete example!

Problem 2.2

Consider the following code, which assumes that the relevant item type is `int`:

```
shared_ptr<Link<int>> a = make_shared<Link<int>>(5, nullptr);
shared_ptr<Link<int>> b = make_shared<Link<int>>(8,
                 make_shared<Link<int>>(13,
                    make_shared<Link<int>>(21, a)));
a->setNext(b);
```

Draw a diagram of memory after those statements are executed, and describe as specifically as you can why that state of affairs can cause problems. Also comment on how you could detect inside a program if this type of thing had happened.

(Hint: think back to Lab 4, and specifically about what happened with the line `shared_ptr<Link<int>> d = c;` and how == worked with pointers.)

Collaboration policy: group work! If you work with other people on this homework, hand in one copy and put all your names on top. There will be a revision cycle for this.

Handing in: On paper, in class on Monday. Just one handin per group, please!