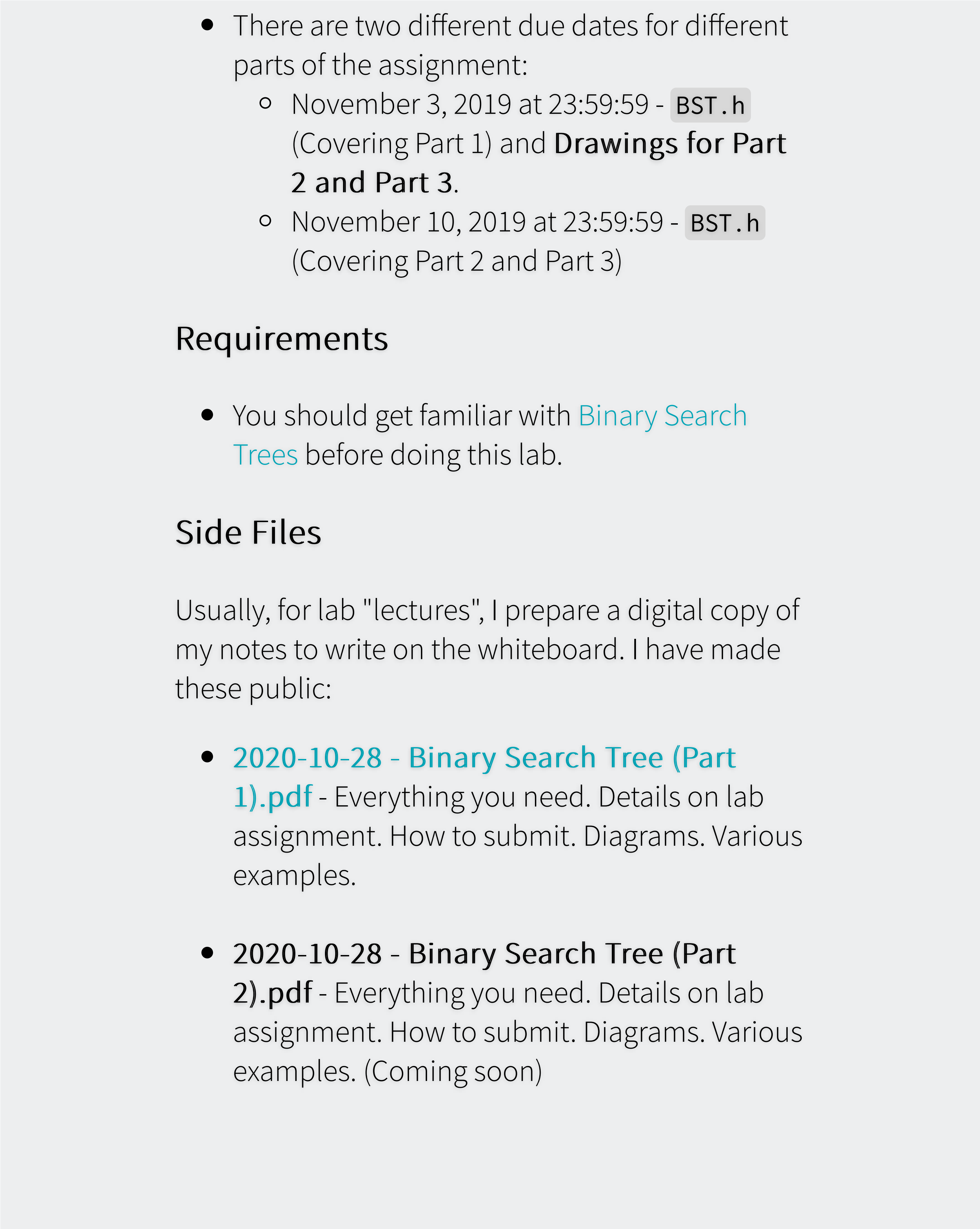
Getting Started

Synopsis

- Read the lab writeup on Instructure Canvas.
- There are **three** parts to this lab. However, you are only making one source file. When you finish the lab, the final part will have the functionality to support all three parts of the assignment.



- There are two different due dates for different parts of the assignment:
    - November 3, 2019 at 23:59:59 - `BST.h` (Covering Part 1) and **Drawings for Part 2 and Part 3**.
    - November 10, 2019 at 23:59:59 - `BST.h` (Covering Part 2 and Part 3)

Requirements

- You should get familiar with Binary Search Trees before doing this lab.

Side Files

Usually, for lab "lectures", I prepare a digital copy of my notes to write on the whiteboard. I have made these public:

- **2020-10-28 - Binary Search Tree (Part 1).pdf** - Everything you need. Details on lab assignment. How to submit. Diagrams. Various examples.

- **2020-10-28 - Binary Search Tree (Part 2).pdf** - Everything you need. Details on lab assignment. How to submit. Diagrams. Various examples. (Coming soon)

Submission

We will be following a variation of these guidelines. I have posted the details on Piazza. However, for archival's sake...

Submission 1

The first submission requires a `BST.h` that supports Part 1, as well as drawings for the next two parts:

```
For 30 points, make simple drawings that illustr
ate how the following functions described below 
are meant
to work, namely, bst::iterator::operator++() whi
ch implements a single step for an inorder trave
rsal, and
bst::lower_bound(key) and bst::upper_bound(key)
which find nodes that bracket a search range for
 a given key,
i.e., lower_bound ≤ key < upper_bound. Add a few
 succinct sentences to explain each drawing. These
are meant for you to think about functionality b
efore you think about code. That is, you can’t c
ode what you
don't understand, and you don't understand what
you can't explain. This makes you understand and
 explain.
```

Canvas lets us view document/image files without even having to download them, which speeds up grading drastically. As such, I have removed the "tar" file restriction for Part 1.

So here's the deal... for **Submission 1**, submit 2 files:

- A **TAR** file containing `BST.h`
- A **document/image** file containing your sketch. If you need more than one file for this, that's fine.

What formats do we take? Since it's a drawing, we'll take image formats such as JPG, PNG, GIF. Concerning document files, we'll take formats such as DOC/DOCX and PDF. **In both cases, we don't care what the filename is, only the file extension.** The only file with a strict name requirement is `BST.h`, inside a TAR, just like all normal code submissions.

If that wasn't clear, behold the following valid submission:

```
lab6_1.tar
drawings.pdf
```

Another valid submission:

```
lab6A.tar
sketch.jpg
```

Files that can't be open (either due to

corruption or other various reasons) will be treated as an invalid submission, resulting in the 25 points for that portion being lost. Failure to open a TAR for code won't result in a 0, but it will result in a 15 point penalty.

Submission 2

This one is much easier. It's the usual.

- BST.h

UNIX COMMAND

```
UNIX> tar -cvf lab6_2.tar BST.h
```

Lab Guide

Unfortunately, I have not written a guide for this lab.