

# Lesson 18- Recursion

⚠ This is a preview of the published version of the quiz

Started: Jul 30 at 1:14pm

## Quiz Instructions

- Listen to the audio and read the PDF of the slides.
- Join the Ohyay workspace today starting at **9:10am**.
- This assignment allows you to work **with a partner if you wish**
- You will each **individually** submit your code and passwords for this **Quiz** to **Canvas**.

## Listen



Bart: Dr. Bart's log, day 6. I'm still trapped down here in AlgoTutorBot's underground facility.

Bart: I've had some issues making progress lately, because all of the doors seem to be locked with some kind of picture-based code.

Bart: I haven't been able to crack the code on my own, but I was able to hack into one of the terminals.

Bart: I started working on a program to analyze the data I found there, in hopes of finding the passcode for these locked doors.

Bart: However, every time I start making progress, another one of those AlgoDeathBot patrols comes by.

Bart: They're not very threatening, but they are very persistent.

Bart: Anyway, besides the frequent interruptions, I'm also starting to run pretty low on food.

Bart: And I don't think I'm any closer to figuring out what is going on down here.

Bart: Why does AlgoTutorBot even need a facility this big? I'm seeing server rooms, laboratories, and even some assembly lines.

Bart: I don't know what's going on here, but I'm concerned I've unleashed something really dangerous on the world.

Bart: ... Okay, you know, while we're on the subject, I probably need to come clean.

Bart: The truth is, I didn't make AlgoTutorBot.

Bart: Here's a protip for anyone who manages to hear this message - if somebody ever emails you out of the blue offering

Bart: an AI that can teach your algorithms course, don't accept it. You'll just end up stuck down in a subterranean complex

Bart: underneath your basement, while you starve to death and dodge killer roombas.

Bart: Speaking of which, I better get moving again, I think I hear another patrol on its way.

Bart: Look, whoever gets this log, I'm attaching the zip file and my code so far.

Bart: Maybe you can help me get the passwords I need, so I can make my way further into the facility.  
Bart: For now, this is Dr. Bart... let's learn about how to survive in an evil underground base, I guess.

## Skim

Here are some slides related to Recursion, that might be helpful. You should also consult other resources, if you are feeling rusty regarding recursion.

[Lesson 18- Recursion.pdf](#)

## Do

Hi students! This is Dr. Bart. I've managed to hack into one AlgoTutorBot's terminals, and I've dumped all the stuff I found into a zip file. I was working on a program that could help me search the files for the security passwords. The defenses around here are really weird, but I'm pretty sure all the information I need is on this drive. Can you help me complete the program to determine the passwords?

## Getting Started

This assignment begins with you downloading a zip file.

**Do NOT extract the zip file.** You will access its contents programatically from within Python.

Download the following two files and make sure they are in the same folder:

1. [find\\_password.py](https://gist.github.com/acbart/ac93f60d0a2139fcd86a16d58a85b81c) [\\_ \(https://gist.github.com/acbart/ac93f60d0a2139fcd86a16d58a85b81c\)](https://gist.github.com/acbart/ac93f60d0a2139fcd86a16d58a85b81c)
2. [mysterious\\_drive.py](https://udel.instructure.com/courses/1563197/files/96497645/download?verifier=Rkc347ciSaVKO1t21w25e3BTpPZkHZ9HdhI2zSNs&download_frd=1) [↓ \(https://udel.instructure.com/courses/1563197/files/96497645/download?verifier=Rkc347ciSaVKO1t21w25e3BTpPZkHZ9HdhI2zSNs&download\\_frd=1\)](https://udel.instructure.com/courses/1563197/files/96497645/download?verifier=Rkc347ciSaVKO1t21w25e3BTpPZkHZ9HdhI2zSNs&download_frd=1)

This assignment requires 4 different kinds of recursion, divided into phases.

You will be editing the `find_password.py` file, filling in the blanks (or in the case of phase 4, writing the body).

## Phase 1) Searching a Zip

First things first, we need to find the security file. I'm pretty confident it's a JSON file, so just do a recursive search of the zipfile looking for the first file that ends with `.json`.

## Phase 2) Accessing JSON

If you haven't seen it before, a JSON file is composed of strings, integers, floats, booleans, `None`s, lists, and objects (aka dictionaries). It's a really convenient format for sharing and storing structured data. And in this case, I'm pretty sure it's holding the security data for every room in this complex. The problem is,

there are so many rooms, and they are organized in a pretty confusing way. There are quadrants, regions, zones, and areas - and their nesting is not consistent.

Fortunately, we just need to find the password for a specific room. So I'll tell you the path of my current location, and you can use a recursive function to access that path down the tree stored in the JSON file.

## Phase 3) Binary Searching Times

We're getting close to the actual password data. Each room has a rotating schedule of passwords. Based on the current hour, we need to get the appropriate maze.

```
[
  {"time": 2, "password": "A"},
  {"time": 5, "password": "B"},
  {"time": 7, "password": "C"},
  {"time": 10, "password": "D"},
]
```

I'll tell you an hour, and you need to use a binary search to efficiently find the appropriate password. For instance, if I said the time was **6**, then you'd want to find the password associated with **5**. If I said it was **7**, then you'd go for the **7**'s password, which is **C**.

Of course, the passwords are not actually going to be **A**, **B**, **C**, etc. That would be too easy. Instead, each password is actually stored in a "String Maze"!

## Phase 4) Solving a 1D Maze







I'll be honest, I didn't get very far with this one. AlgoTutorBot has the weirdest security! It looks like in order to access the password, you have to navigate a 1-Dimensional maze stored in a string. The strings look like this.

→2→9→7↵23X↵→7↵43🐼🐼↵92🌀

Here are the rules of the maze:

- **→** followed by a digit means "go right" that number of spaces.
- **←** followed by a digit means "go left" that number of spaces.
- **↵** followed by two digits means you need to go left, either by the first amount or the second amount.
- **⇒** followed by two digits means you need to go right, either by the first amount or the second amount.
- **⇄** followed by two digits means you need to go left by the first digit spaces, or right by the second digit spaces.
- **X** means that you reached a deadend, and need to return to where you were.
- Any other symbol (e.g., **🐼** or **🌀**) is the password you are seeking.

Let's break down the steps of example maze:

- You always start from the leftmost spot (index 0) in the string.
- At that spot, you will see an arrow followed immediately by a single digit number ( in this case).
- This is telling you to jump two spaces forward, which takes us to the .
- From there, we go to index 11, where we find the .
- Moving forward 7 more spaces takes us to index 18.
- The arrows at 18 are , which means we have to check two possible routes: back 9 and back 2.
- The first route (9 back) takes us to an , so we stop that path.
- The other route (2 back) takes us to an , so that is the password.

In order to avoid retracing your steps unnecessarily, you will have to keep track of visited nodes. In other words, you will implement a recursive depth-first search!

## Submission

When you are done, start this quiz. I will give you a location, a time, and ask you for the password. You should be able to just copy and paste your answers (which will be a single character each) into this quiz. You'll also need to send me your code, so I can see how you did things.

## FAQ

If you are getting issues that there is no member Path in the zipfile module, and you are using Python version 3.7, then you will need to either:

1. Upgrade to Python 3.8 or 3.9
2. Reach out to Dr. Bart to get alternate starter code (essentially replaces Phase 1)

If you are getting issues that "type object is not subscriptable" and you are using Python 3.8, then you will need to either:

1. Upgrade to Python 3.9
2. Include the following at the top of your file: `from __future__ import annotations`
3. Reach out to Dr. Bart to get alternate starter code (essentially removes all the list[str] type annotations, which are strictly optional anyway)
4. Manually remove all the list[str] annotations from the starter file.

If you cannot copy stuff from Canvas, and you need the emoji, all possible ones we might use are here:

<https://gist.github.com/acbart/7eef0f3fd3e8cd5de036c5ff8ce446b5>

(<https://gist.github.com/acbart/7eef0f3fd3e8cd5de036c5ff8ce446b5>)

---

**Question 1**

**2 pts**

```
location = ["Basement 2", "Quadrant A", "Region 1", "North"]  
target_time = 6
```

What is the picture code?

## Question 2

2 pts

```
location = ["SubBasement 3A", "Quadrant A", "East"]  
target_time = 7
```

What is the picture code?

## Question 3

2 pts

```
location = ["Basement 3", "Region 2", "Laboratory 3", "Northeast"]  
target_time = 13
```

What is the picture code?

## Question 4

4 pts

Upload the latest version of your code.

Upload

Choose a File

No new data to save. Last checked at 1:14pm

Submit Quiz