Build your own Internet - Stage B

Computer Networks: Project 2

Current version: **11/28/2020**
Corrections:
None so far

1 Introduction

In this assignment, you will learn how to build and operate a layer-3 network using traditional distributed routing protocols, how different networks managed by different organizations interconnect with each other, and how protocols, configuration, and policy combine in Internet routing.

More specifically, you will first learn how to set up a valid forwarding state within an autonomous system (AS) using OSPF, an intra-domain routing protocol (Stage A). Then, you will learn how to set up valid forwarding state between different ASes, so that an end-host in one AS (e.g., your laptop connected to the university wireless network) can communicate with an end-host in another AS (e.g., Google's server). To do that, you will need to use the only inter-domain routing protocol deployed today: BGP (Stage B). After that, you will implement different BGP policies to reflect business relationships or traffic engineering that exist in the real Internet (Stage C). You will configure both OSPF and BGP through the FRRouting Software Suite, which runs on several software routers assigned to you.

The rest of the document is organized as follows. We first describe the setup you will have to use (Section 2). Then, we list the tasks you should perform (Section 3), submission and other general information (Section 4), and the collaboration and academic integrity policies. We are also providing a separate document giving a crash course on how to configure FRRouting routers.

1.1 Schedule

- ***You can use slip days and submit late for Stages C, but you cannot use slip days or submit late for Preliminary Stage, Stage A, or Stage B--any late submissions for the Preliminary Stage, Stage A, or Stage B will receive a 0.***
- The deadline for Stage B of this project is Dec. 4, at 11pm.

1.2 Collaboration policy

This is an *individual project*, but you can discuss at a conceptual level with other students or consult Internet material, as long as the final code and configuration you submit is completely yours and as long as you do not share code or configuration. Before starting the project, be sure to read the collaboration and academic integrity policy later in this document.

1.3 Instructor/TA information

As explained in the Preliminary Stage document, we will assign each student an autonomous system. Each student is assigned to a TA, as listed on Piazza. Please ask any general questions related to this assignment on Piazza, visible to all unless they reveal private details of your

solution, and only contact your responsible TA for questions that will not be relevant to other students (e.g. you are unable to access your autonomous system).

2. General project setup

See the Preliminary Stage document for further details.

2.1 - Network Topology

2.1.1 Intra-AS topology (same as previous stages, with more details added)

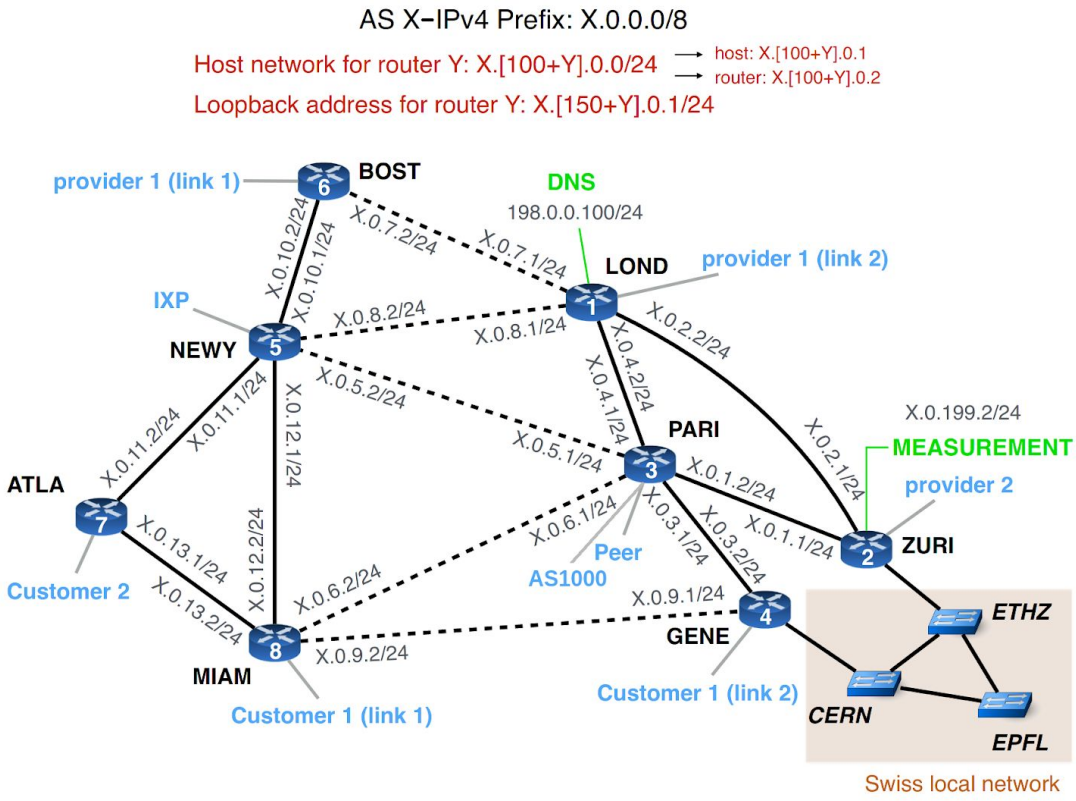
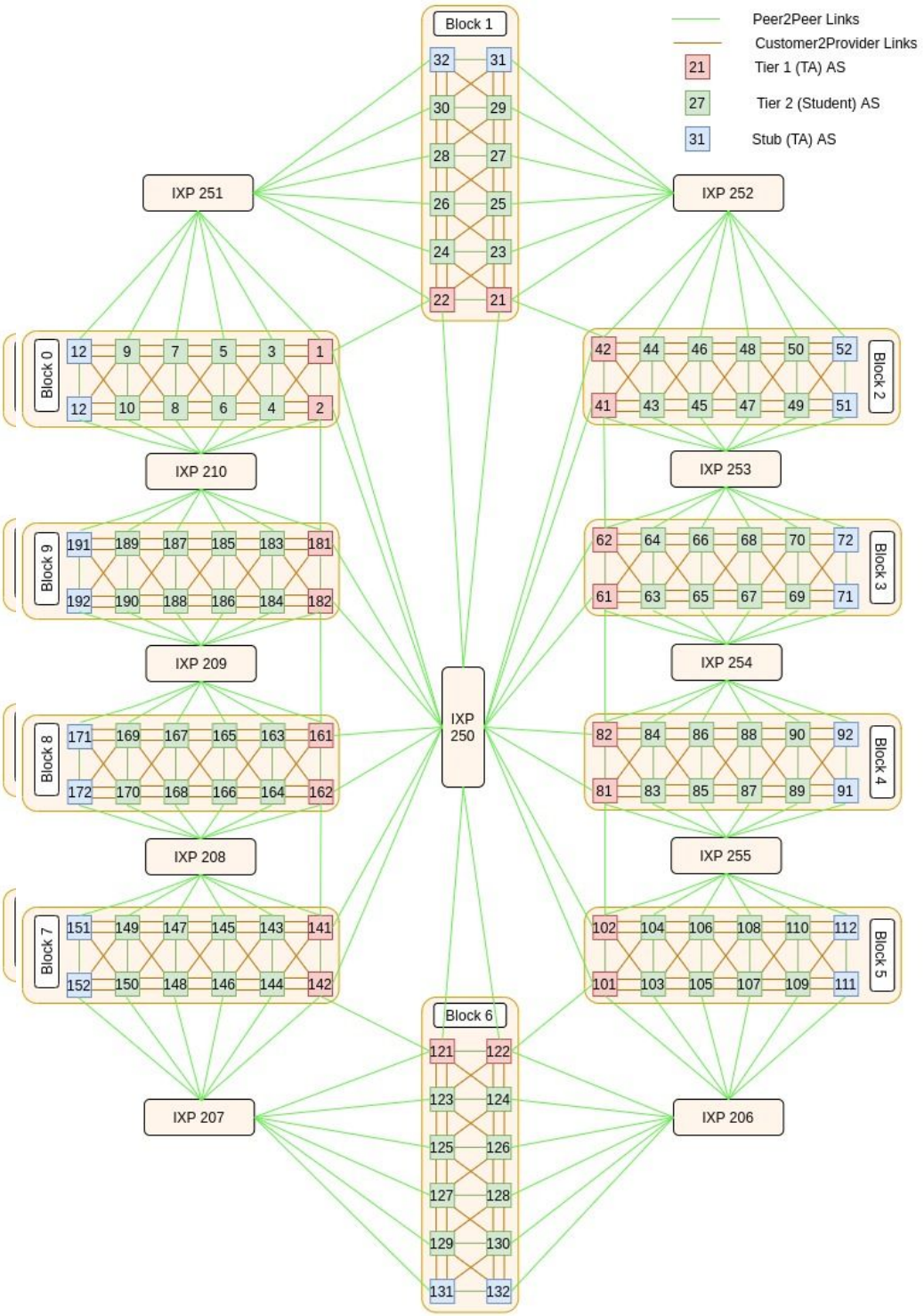


Figure 1: The network topology of your AS. Your AS is composed of 8 routers. A /8 prefix has been assigned to you. You can use it to configure your local networks. The subnets you must use for each of your local networks are indicated on each interface.

The intradomain topology of your network remains the same as described in the Preliminary Stage and Stage A. However, we have added one more detail to the figure:

- Every router has an external connection to one of your neighboring ASes. Some are connected to a provider, some to a customer and others to a peer. NEWY is connected to an Internet eXchange Point (IXP).
    - *PARI has a connection to AS1000 (which you configured in the Preliminary Stage) and to an additional peer, as shown in the updated Figure above.*
- (The fact that some links are solid and some are dotted is not relevant at this time)

2.1.2 Inter-AS Topology



- The class internet is divided into 10 blocks (numbered 0-9)
- There are eleven IXPs within our Internet. IXPs can make it easy to interconnect with many other ASes.
    - One advantage of using an IXP is that an AS can directly peer with another AS through the IXP, instead of reaching it via a provider that it has to pay.
        - The following example illustrates the benefit of being connected to an IXP: AS6 can send traffic to AS187 via the IXP210, instead of paying AS 4 to send the traffic via the path 4-2-181-183-105 if IXP210 is not used.
    - Another advantage is that only one physical connection with an IXP is needed to potentially interconnect with all the other IXP participants. An IXP operates a switch, and multiple ASes connect to the switch. An AS can connect to other ASes across the switch via pairwise (bilateral) BGP sessions, or by a (multilateral) BGP session to a Route Server, which redistributes routes among all participating ASes that connect to it. We will use Route Servers at our IXPs.
    - One IXP is connected to all the Tier1 ASes, allowing them to be connected in a full-mesh fashion. The other IXPs are always interconnecting two blocks. This enables these ASes to peer between them (as long as they respect the BGP customer/provider policies), instead of using (and paying!) their providers.
- Every student AS is connected to five other ASes, and one IXP.
- The neighboring ASes in your block that are closer to the block's Tier 1 ASes are your providers.
    - Eg: For AS 123, ASes 121 and 122 are providers
    - Eg: For AS 108, ASes 105 and 106 are providers.
- The neighboring ASes in your block that are closer to the block's stub ASes are your customers.
    - Eg: For AS 123, ASes 125, 126 are customers.
    - Eg: For AS 108, ASes 109 and 110 are customers.
- The neighboring AS in your block lateral to your position is your peer.
    - Eg: AS 123 and AS 124 are peers.

2.2 The measurement container

- We have set up a measurement container which will enable you to launch traceroute from any Tier2 AS (and not necessarily only your own AS), towards any destination in the class's internet. This will help you to know the paths used towards or through your network. The entire class will share a single measurement container.
- The measurement container is connected to each AS via the interface **`measurement_X`** (where X is your AS number) of the router **`ZURI`**. This interface is preconfigured to use the IP address X.0.199.1/24 (see Fig. 1), where X is your AS number.
    - For example if you are AS 15, your interface **`measurement_15`** at ZURI is preconfigured to use 15.0.199.1/24.

- The X.0.199.1/24 subnet must be reachable from anywhere in your network. You must therefore add it in your OSPF configuration.
- To access the measurement container, use the following command:
  `> ssh -p 2099 root@34.72.133.64`
  The password is posted on Piazza.
- To launch a traceroute, you can use the script **`launch_traceroute.sh`**, which takes as arguments the AS number from which the traceroute starts and the destination IP address (possibly in another AS).
  - For instance,if you want to perform a traceroute from AS 11 to 42.107.0.1 (i.e., the host connected to ATLA in AS 42), just use the following command in the measurement container:

    `> ./launch_traceroute.sh 11 42.107.0.1`

    Note that the traceroute will start from the router ZURI of AS 11, since the measurement container is connected to that router.

2.3 The Connectivity Matrix

For the ease of debugging, we have set up a connectivity matrix that shows you whether an AS can ping another AS. You can access the connectivity matrix from your browser at the following link:

`http://34.72.133.64:2500/matrix.html`

The entry X,Y is the result of AS X pinging AS Y from the matrix interface connected to your PARI router (pre-configured) to the destination AS's ATLA host. The matrix updates ~every 15 minutes, reflecting reachability in the most recent snapshot. Before you set up eBGP sessions, everything will be red. As you set up the eBGP sessions with your neighbors, the matrix will turn green for some pairs of ASes. At the end of this assignment, we hope to see this matrix completely green!

2.4 Looking Glasses

In general, a Looking Glass is a tool that helps us peek into a router's routing information. We have set up a similar server which will enable all the students to look at the BGP routes being received by an AS (not necessarily yours).

To access the Looking Glass, use the following link template:
`http://34.72.133.64:2600/X_[Router]router.txt`

where X is the AS you want to peek into, and [Router] is the name of the router within the AS.

For example, http://34.72.133.64:2600/8_BOSTrouter.txt gives you the output of 'show ip bgp' command run at that particular router.

The information on the web interface is updated every 5 minutes.
Note: The Looking Glass is available to look at Tier2 AS configurations only.

3 Your Tasks

This project is composed of a very short preliminary Stage, then three main Stages (A-C), with Stage B being primarily completed during the class-wide "Internet Hackathon."

- Stage A involves setting up routing within your own network via OSPF and iBGP configuration, and must be finished before the Internet Hackathon.
- Stage B (Internet Hackathon) involves bringing up your eBGP sessions with your neighboring ASes and advertising your prefixes to your neighbors. We will provide details closer to the date of the Hackathon.
- Stage C involves implementing BGP policies according to the business relationships that you have with your neighbors (we will assign the relationships). We will provide details closer to the date of the Hackathon.

Possible plan of attack:

- Familiarize yourself with the previous section, and, using the instructions in the tutorial, access your AS and navigate to routers/hosts.
- For any stage, familiarize with the goals of the stage. Then, refer to the tutorial we provide to find the basics of the commands you will need to enact the goals. Our expectation is that you may need to experiment and try things out to figure out how to accomplish a task based on the guide.

3.1 Preliminary Stage: Your first BGP session (30 points)

Please look at the preliminary stage document for this section of the project.

3.2 Stage A: Configure IGP and iBGP (40 points)

Please look at the Stage A document for this section of the project.

3.3 Stage B: Establish BGP interconnectivity (50 points)

3.3.1 Task - Advertise DNS server and measurement container in OSPF (points awarded as part of Stage A)

Verify that the subnet of the DNS server and the measurement container are visible in OSPF (for instance with show ip route ospf). If they are not, advertise them (it's fine if you didn't advertise them before the Stage A deadline, we will grade you based on whether you do after Stage B). From now on, always prefer to launch traceroute from the hosts because they can use the DNS service (routers cannot). If one host cannot access the DNS server because the OSPF configuration is not ready yet, run traceroute with the option -n so that it does not try to translate each IP address found on the path.

3.3.2 Task - Establish cross-AS links with direct (non-IXP) neighbors (10 points)

Similar to Preliminary Stage Task 3.1.1, establish cross-AS links to all neighboring ASes. You can find your neighbors with the following process:

- Each router connects to one neighboring AS, except PARI (which connects to one PLUS to AS1000) and NEWY (which connects to an IXP).
- Ignoring the IXP connection, the router will have an interface named **`ext_<X>_<CITY>`**, indicating which AS and which router this router should connect to. For example, in AS170, the **`LOND`** router has an interface named **`ext_168_GENE`**. This means that the **`LOND`** router of AS170 should connect to the **`GENE`** router of AS168, which has an interface named an interface named **`ext_170_LOND`.**
- This linked spreadsheet will let you identify who is in charge of an AS, so you can contact the person to establish a link.
- If you contact another AS and do not receive a response within 24 hours, please send a followup email with the following properties:
    - Subject: "followup peering request to ASX from ASY" where X is the other person's ASN and Y is your AS (e.g., "to AS168 from AS170").
    - Include the previous request in the email (forward or attachment)
    - Cc csee4119f20-ta@googlegroups.com
    - Send the followup at least 36 hours before the Stage B submission deadline (so send the followup before Dec 3 at 11am EST, meaning the initial request must be sent by Dec 2 at 11am)

Working with the person in charge of your neighboring AS, establish a cross-AS link. You will have to agree with the person in charge of the other AS as to what subnet you will use and what addresses you will each use within that subnet. You should ***not*** use an address from a /8 assigned to any other AS. You can use otherwise unassigned addresses from a prefix belonging to you and/or your peer, or you can use a subnet from an unallocated or private prefix.

Ping the address taken by your neighbor AS from your peering router to test the connectivity after both ASes have configured their **`ext`** interfaces.

3.3.3 Task - Establish link to IXP (2 point)

Your **`NEWY`** router also connects to IXP **`X`** via an interface named **`ixp_X`**. Configure that interface to use the IP address (and subnet) **`180.X.0.N/24`**, where **`X`** is the IXP number and **`N`** is your ASN.

3.3.4 Task - Configure eBGP sessions (12 points)

Configure the external BGP sessions (eBGP) with your neighboring ASes and with the route server at the IXP. Use the IP address of the neighboring router's interface that connects to your AS (as decided in the earlier task) when specifying a BGP neighbor. As the IXP, establish an eBGP session between your **`NEWY`** router and the IXP's route server, which uses the IP address **`180.X.0.X`**, where **`X`** is the IXP number.

Verify that eBGP sessions are established successfully using the command **`show ip bgp summary`**

3.3.5 Task - Advertise prefix to other ASes (12 points)

Once the eBGP sessions are up, advertise your prefix to your neighboring ASes. You must *only* **originate** the /8 that has been assigned to you (and not other prefixes or /24 sub prefixes of it).

Note: You should announce your BGP announcements to the IXP router server, but, by default, we have configured the route server to not relay your BGP advertisements to its other peers. In a later stage, you will modify your announcements to cause the router server to distribute them.

In the meantime, your peers should advertise to you their /8 prefix, as well as all the /8 prefixes they have learned from other ASes (since there are no BGP policies, yet). Verify that you indeed received those advertisements with **show ip route bgp** or **show ip bgp**, and that those advertisements have been correctly propagated through your iBGP sessions.

Test your connectivity from your hosts towards hosts in other ASes using ping and traceroute.

Important. If you can reach destinations in another AS from a peering router, but not from a non-peering router, you likely have a configuration problem. To fix it, double check that you have added the **`next-hop-self`** command when configuring iBGP sessions. The advertisements from external networks will have the next-hop attribute set to the IP address belonging to the subnet connecting your AS and the other AS, and only one of your routers (the peering one) knows how to reach that address. Thus, this router needs to use **`next-hop-self`** to modify the

next-hop attribute, so that other routers know that they can send traffic to this router if they want to reach the destinations in the external advertisement.

To include in your report: Explain what **`next-hop-self`** does and why you have to use it using an example in your own network. Also, explain on which BGP sessions **`next-hop-self`** is required, why, and why it is not required on other sessions.

Then, show us the results of a **`show ip bgp`** for the router MIAM. You should see the prefixes advertised by your neighboring ASes, which would indicate that your eBGP sessions are correctly configured and that the advertisements are correctly propagated through your iBGP sessions. Then, show us that your neighboring ASes do receive the advertisement for your /8 prefix. To do that, show in your report the result of the Looking Glass for one router located in a neighboring AS. You should see your prefix in the Looking Glass. Finally, show us that you have data-plane connectivity with your neighbors by including in your report the result of a traceroute from your **`PARI-host`** to the **`PARI-host`** of one of your neighboring ASes.

3.3.5 Task - List eBGP configuration (14 points)

Please generate a new **`as<X>-connections.txt`** ASCII file describing the inter-AS connections you established with other networks (including AS1000 and the IXP). In place of ***<X>***, please put the number of your AS (for example, **`as5-connections.txt`**). Below we describe how to submit the file. The file format should be as follows:

local_AS	remote_AS	local_peering_city	subnet	local_ip	remote_ip

Table 2: An example of the headers of the new table

3.4 Stage C: BGP policy & delivering Internet content (80 points)

Further details of Stage C will be released after the Hackathon.

4 Submission and other information

The project is worth 200 points (Preliminary Stage: 30 points, Stage A: 60 points, Stage B: 50 points, Stage C: 60 points). In reports that you submit, clearly label which question you are answering with your answer/screenshot by writing the Task number and name, e.g., “4.1.3 Task - Advertise prefix to AS1000.”

Remember to run ./save_configs.sh, copy the corresponding file generated to the machine you plan to submit from, and rename the folder before submitting to make sure you submit the version you intend.

You will submit it via Courseworks. The format of your submission is the same as your preliminary stage.
You must include the following files in a compressed file called **project2b_Lastname_Firstname_ASN.zip** (for example, *project2b_Katz-Bassett_Ethan_AS10.zip* if the professor were AS10 in the topology). The files are:

1. Written report, with filename **report.pdf**
2. A new ASCII file called **`as<X>-connections.txt`**, using the format described in Task - List eBGP configuration. In place of ***<X>***, please put the number of your as (for example, **`as10-connections.txt`**).
3. The entire saved **configs[date][time]** directory on your AS under home directory renamed as **configs**.

The result of the above will be a list of text files named as **routers (PARI.txt, NEWY.txt, ATLA.txt etc) or Switches (ETHZ.txt, CERN.txt, EPFL.txt)**. Zip this folder along with your report.pdf. Your zipped submission file, for example, should have the following directory structure **after being unzipped** (the top level must contain the project2a* directory):

project2b_Venkatagiri_Shiv_AS7/
├──── report.pdf
├──── configs
│ ├──── ATLA.txt
│ └──── PARI.txt
│ ├──── .
│ ├──── .
│ ├──── .

5 Academic integrity: Zero tolerance on plagiarism

The rules for Columbia University, the CS Department, and the EE Department (via SEAS: 1 and 2) apply. It is your responsibility to carefully read these policies and ask the professor (via Piazza) if you have any questions about academic integrity. **Please ask the professor before submitting the assignment, with enough time to resolve the issue before the deadline**. A misunderstanding of university or class policies is not an excuse for violating a policy.

This class requires closely obeying the policy on academic integrity, and has zero tolerance on plagiarism for all assignments, including both projects/programming assignments and written assignments. By zero tolerance, we mean that the minimum punishment for plagiarism/cheating is a 0 for the assignment, and all cases will be referred to the Dean of Students.

This assignment must be completed individually. For programming assignments, in particular, you must write all the code you hand in yourself, except for code that we give you as part of the

assignments. You are not allowed to look at anyone else's solution (including solutions on the Internet, if there are any), and you are not allowed to look at code from previous years or ask people who took the class in previous years for help. You may discuss the assignments with other students at the conceptual level, but you may not write pseudocode together, or look at or copy each other's code. Helping other students violate the policy (for example, letting them look at your code) is a violation, even if you completed the code yourself. Please do not publish your code or make it available to future students -- for example, please do not make your code visible on Github. Uploading course materials to sites such as CourseHero, Chegg or Github is academic misconduct at Columbia (see pg 10).

You may look at documentation from the tools' websites. However, you may not use external libraries or any online code unless granted explicit permission by the professor or TA. For written (non-programming) answers, if you quote material from textbooks, journal articles, manuals, etc., you **must** include a citation that gives proper credit to the source to avoid suspicion of plagiarism. If you are unsure how to properly cite, you can use the web to find references on scientific citations, or ask fellow students and TAs on Piazza.

Deliberately disrupting shared project infrastructure or the ability of others to use shared infrastructure is a serious violation. If you are unsure how to use shared infrastructure, please ask classmates or instructors first.

For each programming assignment, we will use software to check for plagiarized code. **So, be really careful and do not read or copy code or text.**

Note: You must set permissions on any homework assignments so that they are readable only by you. You may get reprimanded for facilitating cheating if you do not follow this rule.