CS 168 Spring 2025 — Introduction to the Internet Final Exam

Solutions last updated: Saturday, May 17, 2025

Print Your Name: _____

Print Your Student ID: _____

You have 170 minutes. There are 8 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	7	8	Total
Points:	20	15	10	11	10	12	14	8	100

For questions with **circular bubbles**, you may select only one choice.

Ⓐ Unselected option (Completely unfilled)

Ⓑ Don't do this (it will be graded as incorrect)

● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select one or more choices.

■ You can select

■ multiple squares

☑ Don't do this (it will be graded as incorrect)

Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

Honor Code: Read the honor code below and sign your name.

I understand that I may not collaborate with anyone else on this exam, or cheat in any way. I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct and may further result in, at minimum, negative points on the exam.

Sign your name: _____

Clarifications

- Q6 - The top of page 12 should say “there are k ~~switches~~ **servers** in the middle layer.”

Q1 *Potpourri* **(20 points)**

For the next two subparts: Recall that in Project 3 (Transport), the starter code sets `self.snd.wnd = self.TX_DATA_MAX`, where `self.TX_DATA_MAX` is some large hard-coded constant. In Stage 5, you updated this code to change how the window size is set.

Q1.1 (3 points) If we didn't implement Stage 5, and instead left the starter code unchanged, what would happen in the resulting TCP implementation? Select all that apply.

- [A] The receiver might be overwhelmed with too many out-of-order packets.
- [B] The network might be overwhelmed with too many packets.
- [C] The sender would always have to wait for packet i to be acked before sending packet $i + 1$.
- (D) None of the above

Solution:

`self.snd.wnd`, the window size, is supposed to implement flow control (avoid overwhelming receiver) and congestion control (avoid overwhelming network).

If we always hard-code the window size to some large number, then the receiver and the network may be overwhelmed.

The last option is false. Waiting for i to be acked before sending $i + 1$ is the behavior when the window is hard-coded to 1 packet, but in this implementation, we're hard-coding the window to some large constant value instead.

Q1.2 (2 points) In Stage 5, what did you set `self.snd.wnd` to, and why?

- (A) A value reported from the other side, for flow control.
- (B) A value reported from the other side, for congestion control.
- (C) A value computed by your code, for flow control.
- (D) A value computed by your code, for congestion control.

Solution:

Project 3 (Transport) did not implement congestion control, so those options are false.

You set `self.snd.wnd` to implement flow control, and flow control is implemented by having the other side report how much buffer space it has left.

Q1.3 (2 points) What does a TCP receiver do when it receives 2 identical duplicate packets, both with sequence number 50?

Ⓐ Send one packet with ack number 50.
Ⓒ Send one packet with ack number 51.
Ⓑ Send two packets with ack number 50.
● Ⓓ Send two packets with ack number 51.

Solution:

The correct ack number is 51, because that is the next byte that we expect to receive, aka the first unreceived byte.

When we receive two duplicate packets, we must send back two acks. The first ack might have been dropped, which caused the packet to be re-transmitted. In that case, we must reply with a second ack to help the sender know that the packet was received.

Q1.4 (1 point) When deploying CDNs, each CDN server must contain a complete copy of all the resources on the origin server.

Ⓐ True
● Ⓑ False

Solution:

False. A CDN server could have just a subset of the resources (e.g. the most highly-requested resources), such that some resources are served by the CDN, and other resources (e.g. less-requested ones) are served by the origin server.

Q1.5 (1 point) Which HTTP header helps ensure that the cache keeps up-to-date copies of cached resources?

● Ⓐ Cache-Control
Ⓑ Content-Type
Ⓒ Version
Ⓓ User-Agent

Solution:

The Cache-Control header indicates when the cached data expires (and needs to be requested again). This ensures that stale data eventually expires, and the user eventually requests a more up-to-date copy of the data.

Q1.6 (1 point) To access a **private** HTTP cache, the user must send at least one packet through the network.

Ⓐ True
● Ⓑ False

Solution:

False. The private cache can be in the user's browser on their own computer. No TCP connection is needed to access data on the user's own computer.

Q1.7 (1 point) To access a **proxy** HTTP cache, the user must send at least one packet through the network.

(A) True (B) False

Solution:

True. The proxy cache lives in the network, so the user must open a TCP connection to access the proxy cache.

Q1.8 (1 point) How many physical DNS root name servers exist?

(A) 1 (B) 2 (C) 13 (D) More than 13

Solution:

There are 13 root-server domains that all use anycast, so thousands of servers share their IPs.

Q1.9 (1 point) Consider a router hashing each connection's 5-tuple to decide how to forward the packet.

If there are two shortest paths to a given destination, this router guarantees that exactly half of the packets take one path, and half of the packets take the other path.

(A) True (B) False

Solution:

False. For example, consider a mice flow hashed to one path, and an elephant flow hashed to another path.

In the next two subparts, consider a datacenter using overlay and underlay networks to forward packets. Server Y is some arbitrary server somewhere in the datacenter.

Q1.10 (2 points) What IP addresses can be found in the forwarding table(s) of a **virtual switch** on Server Y? Select all that apply.

- [A] The physical IP of Server Y.
- [B] Physical IPs of some other servers.
- [C] The virtual IP of all VMs on Server Y.
- [D] Virtual IPs of VMs on some other servers.
- (E) None of the above

Solution:

The virtual switch needs an entry that decapsulates packets where the destination is the physical IP of Server Y.

The virtual switch must also know about the physical IPs of other servers, to fill out its underlay forwarding table with next-hops to those servers.

The virtual switch must also know about the virtual IPs of VMs on its own server, so that it can forward decapsulated packets to the appropriate VM.

The virtual switch needs to know about the virtual IPs of other VMs. When a VM on Server Y sends a packet to the virtual switch, it only has an overlay header, and the VM must convert the overlay destination (virtual IP on some other server) to the corresponding underlay IP address. This allows the virtual switch to then encapsulate the packet with the underlay destination IP.

Q1.11 (2 points) What IP addresses can be found in the forwarding table(s) of a **physical router** directly connected to Server Y? Select all that apply.

- [A] The physical IP of Server Y.
- [B] Physical IPs of some other servers.
- [C] The virtual IP of all VMs on Server Y.
- [D] Virtual IPs of VMs on some other servers.
- (E) None of the above

Solution:

The router operates in the underlay, so it knows about physical IPs. The router knows the physical IP of Server Y so that it can forward packets to this server. Also, the router knows the physical IP of other servers so that it can populate its underlay forwarding table with next-hops to those servers.

The router does not need to know about any overlay virtual IPs, since it only processes the outer header of an encapsulated packet, and never the inner header.

Q1.12 (1 point) In software-defined networking, an OpenFlow table can be used to program destination-based forwarding rules into a router.

(A) True (B) False

Solution:

True. OpenFlow tables can be used to define more complex rules (e.g. traffic engineering), but they can also be used for simple rules like destination-based forwarding.

Q1.13 (1 point) In a network with low-bandwidth links, offloading operations to the network interface card (NIC) will result in significant performance benefits.

(A) True (B) False

Solution:

False. Offloading is beneficial when the network is so high-performance that the operations at the hosts are the bottleneck.

If the network is low-performance, then the network itself is the bottleneck, and improving performance at the hosts won't result in significant performance benefits.

Q1.14 (1 point) RDMA uses the operating system's TCP implementation to ensure reliability.

(A) True (B) False

Solution:

False. The operating system implements TCP in software, and RDMA is designed to transfer data in hardware, with minimal involvement from software.

Q2 *TCP Congestion Control* **(15 points)**

In this question, you are sending data using the TCP congestion control algorithm seen in lecture.

Assumptions for the entire question:
- All TCP values are measured in packets (not bytes), unless otherwise specified.
- The sender always has new data to send, and RWND is very large.

Q2.1 (1 point) Immediately after the TCP handshake, suppose you set CWND = 5 packets.

In general (not necessarily for this flow), why might senders initialize CWND to 5, instead of 1?

Ⓐ To prevent cheating.

Ⓑ To distinguish congestion and corruption.

Ⓒ To improve performance for short flows.

Ⓓ To improve performance for long flows.

Solution:

If CWND is initialized to 1, then a short flow with just a few packets would take unnecessarily long to complete, since the window is still growing by the time the connection is complete.

By initializing CWND to a higher number like 5, a short flow can send most (or all) of its packets right away.

Initializing CWND to a higher number doesn't help to prevent cheating, and it doesn't help distinguish congestion and corruption. For long flows, only the first few packets are affected by the higher initial value of CWND, and the performance for the rest of the connection doesn't change, so the overall performance impact is negligible.

Immediately after the TCP handshake, you are in Slow Start mode, with the following settings:
- CWND = 5, SSTHRESH = ∞
- The sender's first data packet has sequence number #20.

Q2.2 (2 points) Suppose you receive an ack for packet #20 (i.e. the receiver has received packet #20, and sends you an ack).

At this point, which packets are allowed to be in flight? Select all that apply.

☐ A #20	☐ C #22	☐ E #24	☐ G #26	☐ I #28	☐ K #30
☐ B #21	☐ D #23	☐ F #25	☐ H #27	☐ J #29	☐ L #31

Solution:

The first un-acked packet is #21, so this is the start of our window.

The window size was 5, but in slow-start mode, each new ack increases our window by 1. The ack for #20 has increased our window to $5 + 1 = 6$.

Our window starts at #21 and is 6 packets large, so it allows us to send #21 through #26, inclusive.

Q2.3 (2 points) Suppose you then receive an ack for packet #21.

At this point, which packets are allowed to be in flight? Select all that apply.

[A] #20	[C] #22	[E] #24	[G] #26	[I] #28	[K] #30
[B] #21	[D] #23	[F] #25	[H] #27	[J] #29	[L] #31

Solution:

The reasoning is similar to the previous subpart.

The first un-acked packet is #22, so this is the start of our window.

The window size was 6, but in slow-start mode, each new ack increases our window by 1. The ack for #21 has increased our window to $6 + 1 = 7$.

Our window starts at #22 and is 7 packets large, so it allows us to send #22 through #28, inclusive.

The rest of this question is independent of earlier subparts. Each subpart continues on from the previous one.

Some time later, you are in Slow Start mode, with the following settings:

- CWND = 14, SSTHRESH = ∞
- All packets up to, and including packet #30, have been sent and acked.
- Packets #31 through #44, inclusive, have been sent, but not acked.
- All packets #45 and later have not been sent.

Suppose that packet #31 is dropped in transit. All other packets are successfully sent, and the receiver sends you acks for packets #32, #33, #34, etc., in that order, with no timeouts.

Q2.4 (1 point) In this scenario, TCP will switch from Slow Start mode to Fast Recovery mode.

TCP switches to Fast Recovery mode immediately after you receive the ack corresponding to which packet?

Ⓐ #32 Ⓑ #33 Ⓒ #34 Ⓓ #35 Ⓔ #36 Ⓕ #37

Solution:

You would receive 3 duplicate acks, causing you to transition into Fast Recovery mode.

The 3 duplicate acks correspond to packets #32, #33, and #34 being acked.

Q2.5 (2 points) What is the value of CWND the instant **after** TCP switches from Slow Start mode to Fast Recovery mode?

Reminder: During Slow Start mode, duplicate acks do not change the value of CWND.

Ⓐ 7 Ⓑ 8 Ⓒ 9 Ⓓ 10 Ⓔ 11 Ⓕ 12

Solution:

Once you receive 3 duplicate acks, CWND will be halved from 14 to 7.

Then, fast recovery will temporarily give credit for the 3 acked packets (corresponding to the 3 duplicate acks), so we'll increase CWND by 3, to $7 + 3 = 10$.

Reminder: You are now in Fast Recovery mode.

Q2.6 (2 points) The receiver receives packets in this order: #32, #33, #34, ..., #43, #44, #31 (retransmitted).

The receiver sends acks for these packets, and you receive these acks in order, with no timeouts.

What is the value of CWND the instant **before** TCP switches out of Fast Recovery mode?

Ⓐ 7 Ⓑ 10 Ⓒ 13 Ⓓ 15 Ⓔ 17 **Ⓕ 20**

Solution:

During fast recovery, each ack increases CWND by 1.

The acks for #32, #33, #34 were the three duplicate acks that caused you to switch into Fast Recovery mode, and you already gave credit for these acks by increasing CWND from 7 to 10.

Then, the acks for #35, #36, #37, ..., #43, #44 increase CWND by 10, from 10 to 20.

(Writers' note: The answer choices are separated out a bit to avoid off-by-one answers like 19 or 21.)

Q2.7 (2 points) What is the value of CWND the instant **after** TCP switches out of Fast Recovery mode?

Ⓐ 7 Ⓑ 10 Ⓒ 13 Ⓓ 15 Ⓔ 17 Ⓕ 20

Solution:

When we left Slow Start mode, the intended behavior was to decrease CWND from 14 to 7.

Fast Recovery temporarily increased CWND, but once we leave Fast Recovery mode, we should go back to the originally-intended value of CWND = 7.

Q2.8 (2 points) At the instant **before** TCP switches out of Fast Recovery mode, which packets have been sent out?

All packets up to, and including _____, have been sent out.

Ⓐ #38 Ⓑ #41 Ⓒ #44 Ⓓ #47 **Ⓔ #50** Ⓕ #53

Solution:

Before you leave Fast Recovery, you still haven't received an ack for #31 yet, so the left side of the window is #31.

CWND has increased to 20 during Fast Recovery, so this creates a window including packets #31 to #50, inclusive.

Q2.9 (1 point) After TCP switches out of Fast Recovery mode, what mode is TCP in?

Ⓐ Slow Start Ⓑ Congestion Avoidance

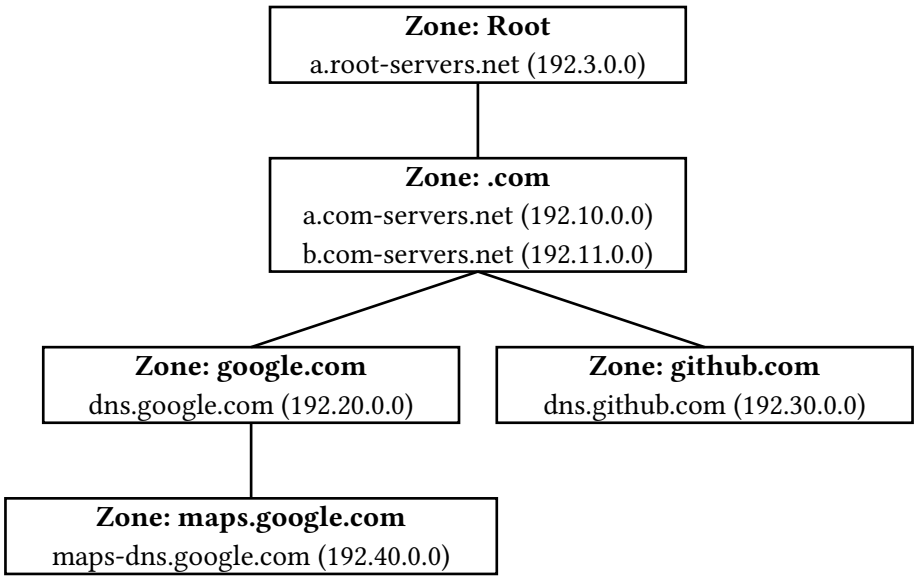
Solution:

Once you receive the ack for #31 (which is a new ack, not a duplicate ack), you can leave Fast Recovery mode and enter Congestion Avoidance mode.

We would not re-enter Slow Start mode, since that only happens on a timeout.

Q3 *DNS* (10 points)

Consider the following DNS name server hierarchy. Each box represents a zone. Each box contains the domains and corresponding IP addresses for all name servers that are authoritative for that zone. Assume that no other zones exist, besides the ones shown.



For each of the following records (name, type, value), select the zone that could provide the given record.

If the given record is invalid, or no zone would provide the given record, select "None."

Q3.1 (1 point) `classroom.github.com A 142.250.9.138`

Ⓐ Root Ⓒ google.com Ⓔ github.com

Ⓑ .com Ⓓ maps.google.com Ⓕ None

Solution:

This record provides the IP address of `classroom.github.com`, which is a subdomain in the `github.com` zone.

Q3.2 (1 point) `google.com NS dns.google.com`

Ⓐ Root Ⓒ google.com Ⓔ github.com

Ⓑ .com Ⓓ maps.google.com Ⓕ None

Solution:

This record allows the .com zone to delegate authority to the google.com zone, by saying that dns.google.com is an authoritative name server for the google.com zone.

Q3.3 (1 point) `google.com NS maps-dns.google.com`

Ⓐ Root
Ⓒ google.com
Ⓔ github.com
Ⓑ .com
Ⓓ maps.google.com
● Ⓕ None

Solution:

This record says that the maps-dns.google.com name server is authoritative for the google.com zone, but this is false. (This name server is authoritative for the maps.google.com zone.) Therefore, this record will not be sent by any of the name servers.

Q3.4 (1 point) `maps-dns.google.com A 192.40.0.0`

Ⓐ Root
● Ⓒ google.com
Ⓔ github.com
Ⓑ .com
Ⓓ maps.google.com
Ⓕ None

Solution:

This additional (glue) record can be sent alongside a record like `maps.google.com NS maps-dns.google.com`, to help a client learn the IP address of the maps.google.com name server.

In the next three subparts, a modification is given. Select the zone that needs to update its record(s) to process this modification. If no records need to be updated, select "None."

Q3.5 (2 points) The IP address of images.google.com is changed from 142.250.189.14 to 142.250.189.15.

(A) Root
(B) .com
(C) google.com
(D) maps.google.com
(E) github.com
(F) None

Solution:

This IP address belongs to the google.com zone, so the google.com zone must update its records for this changed IP address.

Q3.6 (2 points) A new name server, c.com-servers.net (192.12.0.0), is added. This name server is authoritative for the .com zone.

(A) Root
(B) .com
(C) google.com
(D) maps.google.com
(E) github.com
(F) None

Solution:

The root zone must update its records to add new NS and A records for this new name server.

Q3.7 (2 points) A new mirror name server for the .com zone is installed. Using anycast, this new name server has IP address 192.10.0.0.

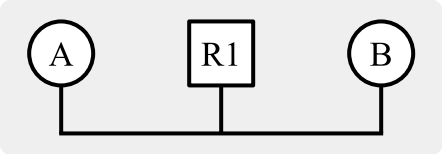
(A) Root
(B) .com
(C) google.com
(D) maps.google.com
(E) github.com
(F) None

Solution:

None of the existing records need to change. Anycast allows multiple mirror name servers to be represented using the same domain name and IP address.

Q4 *End-to-End* **(11 points)**

Consider a subnet with router R1, and hosts A and B, all connected on a single shared medium (also called a bus).



In this question, each subpart continues on from the previous one.

R1 is using NAT with Port Address Translation, as shown in lecture, with the following addresses:

- R1's public address is 143.3.4.5, and its private address is 192.168.1.1.
- R1 can allocate private addresses 192.168.1.2, 192.168.1.3, and so on, to hosts.
- The DNS recursive resolver has IP 8.8.8.8.

Q4.1 (2 points) Host A joins the network, with all caches empty and no active connections.

In Host A's DHCP Discover request, what is the destination IP address?

Ⓐ 8.8.8.8 Ⓑ 255.255.255.255 Ⓒ 143.3.4.5 Ⓓ 192.168.1.1

Solution:

The DHCP Discover request must be broadcast, because Host A doesn't have any information in its cache when it first joins the network.

The broadcast IP address is the all-ones address, or 255.255.255.255. (The all-zeroes address would be used as the source IP address, since the new host doesn't have its own IP address yet.)

Q4.2 (2 points) In R1's DHCP Offer to Host A, what is the subnet in the offer?

Ⓐ 0.0.0.0/0 Ⓑ 192.168.1.0/24 Ⓒ 192.168.1.1/32 Ⓓ 143.3.4.0/24

Solution:

The range 192.168.1.0/24 includes the IP addresses 192.168.1.1, 192.168.1.2, 192.168.1.3, and so on. These are the IP addresses that R1 will assign in this subnet.

At this point, Host A has completed the DHCP handshake, and has been assigned IP address 192.168.1.2. Host A has not initiated any connections yet.

Host A types www.berkeley.edu in their browser. In the next 3 subparts, fill out the fields of the first IP packet that Host A creates and sends as a result.

Q4.3 (1 point) Source IP:

Ⓐ 8.8.8.8 Ⓑ 143.3.4.5 Ⓒ 192.168.1.1 Ⓓ 192.168.1.2

Solution:

Host A is assigned IP address 192.168.1.2, so this is the source IP address for packets sent by Host A.

Q4.4 (1 point) Destination IP:

(A) 8.8.8.8 (B) 143.3.4.5 (C) 192.168.1.1 (D) 192.168.1.2

Solution:

Host A must first send a DNS request to learn the IP address of www.berkeley.edu.

Q4.5 (2 points) Destination MAC:

(A) A's MAC (C) R1's MAC (E) berkeley.edu's MAC

(B) B's MAC (D) DNS server's MAC (F) `ff:ff:ff:ff:ff:ff`

Solution:

The next-hop for the DNS request is R1.

The first IP packet that Host A creates eventually reaches R1. At this point, R1 possibly rewrites some of the header fields, and then forwards the packet. In the next two subparts, fill out the fields of the packet that R1 sends out (possibly with rewritten headers).

Q4.6 (1 point) Source IP:

(A) 8.8.8.8 (B) 143.3.4.5 (C) 192.168.1.1 (D) 192.168.1.2

Solution:

Because NAT is enabled, R1 must replace the private source IP (192.168.1.2) with the public IP address (143.3.4.5).

Q4.7 (1 point) Destination IP:

(A) 8.8.8.8 (B) 143.3.4.5 (C) 192.168.1.1 (D) 192.168.1.2

Solution:

The destination is unchanged by NAT rewriting the source IP.

Q4.8 (1 point) Will the IP address 192.168.1.2 always be associated with Host A?

(A) Yes (B) No

Solution:

Host A is only given a temporary lease on the IP address 192.168.1.2. Eventually, when the lease expires, this IP address may be allocated to a different host.

Q5 *STP* **(10 points)**

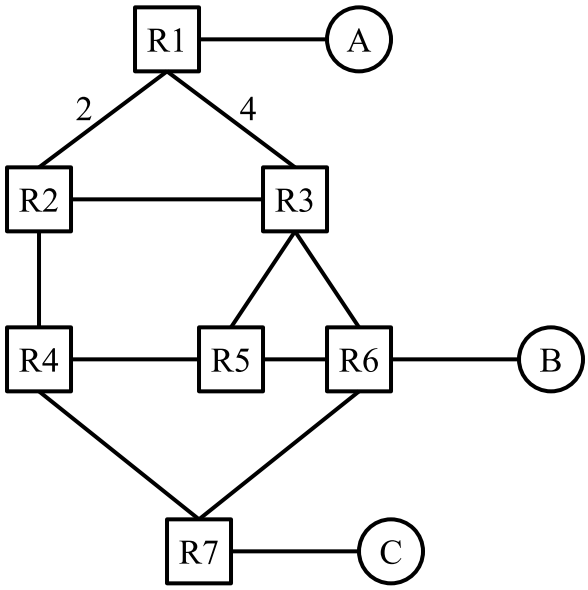
Consider running the Spanning Tree Protocol (STP) for the network topology to the right.

Assume the IDs are ordered according to the router labels. For example, R4 has a lower ID than R5.

Assume the links with no label have a cost of 1.

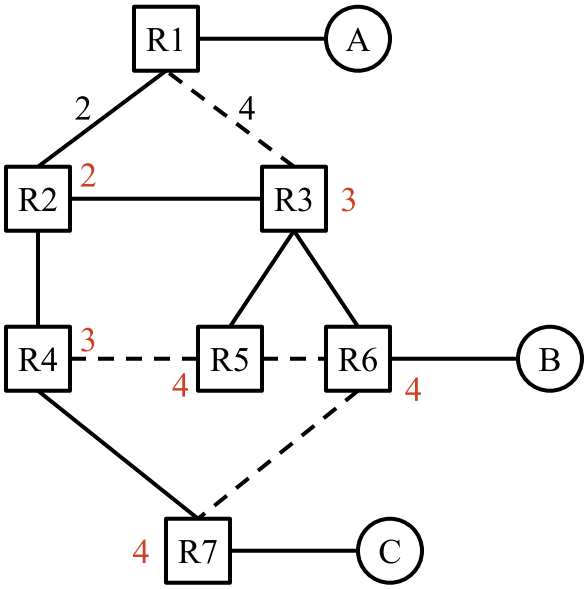
For each of the next six subparts, select the link disabled by the given router. Option "R1" means "the link to R1," and likewise for other options.

If the router does not disable any link, select "None."



Solution:

For the next 6 subparts, see the solution diagram below. The red numbers denote the cost from each router to the root (R1).



Q5.1 (1 point) Which link (if any) does R2 disable?

(A) None (B) R1 (C) R3 (D) R4

Solution:

R1 is the best (and only) path toward the root, so don't disable the link (root port).

R3 is further away, so don't disable the link (designated port).

R4 is further away, so don't disable the link (designated port).

Q5.2 (1 point) Which link (if any) does R3 disable?

(A) None (B) R1 (C) R2 (D) R5 (E) R6

Solution:

R1 points toward the root, but it is not the best path toward the root (cost 4), so disable this link (blocked port).

R2 is the best path toward the root (cost 3), so don't disable the link (root port).

R5 is further away, so don't disable the link (designated port).

R6 is further away, so don't disable the link (designated port).

Q5.3 (1 point) Which link (if any) does R4 disable?

(A) None (B) R2 (C) R5 (D) R7

Solution:

R2 is the best (and only) path toward the root, so don't disable the link (root port).

R5 is further away, so don't disable the link (designated port).

R7 is further away, so don't disable the link (designated port).

Q5.4 (1 point) Which link (if any) does R5 disable?

(A) None (B) R3 (C) R4 (D) R6

Solution:

R3 is the best path toward the root (cost 4, next-hop ID is R3 which is lower), so don't disable the link (root port).

R4 points toward the root, but it is not the best path toward the root (cost 4, next-hop ID is R4 which is higher), so disable this link (blocked port).

R6 is further away, so don't disable the link (designated port).

Q5.5 (1 point) Which link (if any) does R6 disable?

Ⓐ None Ⓑ R3 Ⓒ R5 Ⓓ R7

Solution:

R3 is the best path toward the root (cost 4), so don't disable the link (root port).

R5 points toward the root (same cost, but lower ID), but it is not the best path toward the root (cost 5), so disable this link (blocked port).

R7 is further away (same cost, but higher ID), so don't disable the link (designated port).

Q5.6 (1 point) Which link (if any) does R7 disable?

Ⓐ None Ⓑ R4 Ⓒ R6

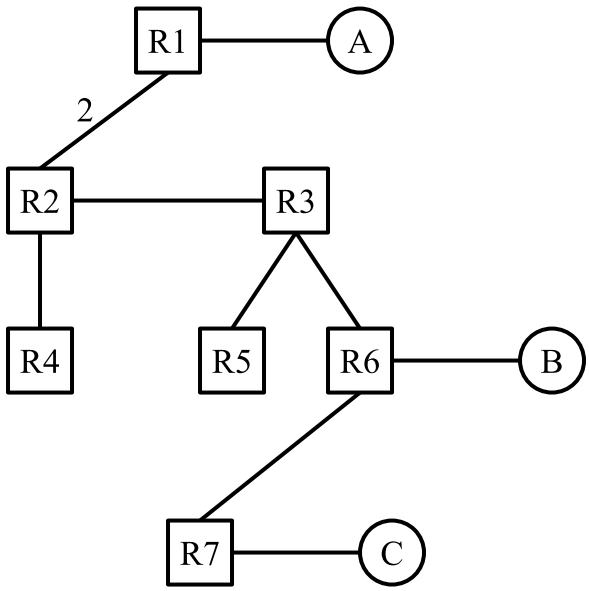
Solution:

R4 is the best path toward the root (cost 4), so don't disable the link (root port).

R6 points toward the root (same cost, but lower ID), but it is not the best path toward the root (cost 5), so disable this link (blocked port).

Suppose STP has converged. Regardless of your answers to the previous subparts, assume that the following 4 links are disabled (also shown in the diagram to the right):

- R1-to-R3
- R4-to-R5
- R5-to-R6
- R4-to-R7



Switches R1 to R7 are all learning switches.

All forwarding tables start out empty.

In each of the next two subparts, select all switches that will receive the given packet.

The packets are sent one after the other. In other words, forwarding table entries created in one subpart carry over to later subparts.

Q5.7 (2 points) A sends a packet to C.

A R1 B R2 C R3 D R4 E R5 F R6 G R7

Solution:

All forwarding tables are empty, so every switch floods the packet to all neighbors.

Also, every switch now has an entry for the next-hop to A.

Q5.8 (2 points) C sends a packet to A.

A R1 B R2 C R3 D R4 E R5 F R6 G R7

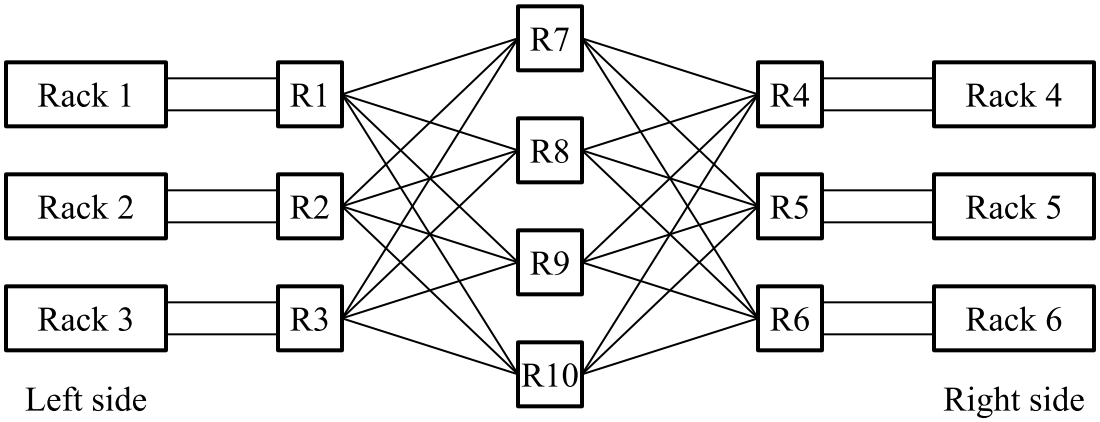
Solution:

Since every switch now has an entry for the next-hop to A, this packet will be forwarded along the path to A (C—R7—R6—R3—R2—R1—A), with no flooding needed.

Q6 *Datacenters* **(12 points)**

Consider the Clos-like topology below.

- Each rack has 2 servers, and each server has its own link to the adjacent router. In other words, each rack has two links to its adjacent router.
- The bandwidth of each link, and the line rate of each server, are all equal.
- Racks 1, 2, 3 (with 6 servers in total) are on the left side. Racks 4, 5, 6 (with 6 servers in total) are on the right side.



Q6.1 (1 point) Each server on the left side can send data to a corresponding server on the right side, at full line rate.

(In other words: We can create 6 connections, each sending at full line rate, where the 6 left-side servers each participate in one connection, and the 6 right-side servers each participate in one connection.)

Ⓐ True Ⓑ False

Solution:

True. We can draw a dedicated path from each left-side server to a single corresponding right-side server. Since each pair of servers has its own dedicated path, the servers can all communicate at full line rate.

Rack 1, Server 1 — R1 — R7 — R4 — Rack 4, Server 1

Rack 1, Server 2 — R1 — R8 — R4 — Rack 4, Server 2

Rack 2, Server 1 — R2 — R7 — R5 — Rack 5, Server 1

Rack 2, Server 2 — R2 — R8 — R5 — Rack 5, Server 2

Rack 3, Server 1 — R3 — R7 — R6 — Rack 6, Server 1

Rack 3, Server 2 — R3 — R8 — R6 — Rack 6, Server 2

Q6.2 (1 point) For this subpart only, suppose each rack has 3 servers, instead of 2 servers. Each of the 3 servers still has its own link to the adjacent router.

Each server on the left side can send data to a corresponding server on the right side, at full line rate.

(A) True (B) False

Solution:

True. As in the previous subpart, we can still draw a dedicated path from each left-side server to a single corresponding right-side server. Since each pair of servers has its own dedicated path, the servers can all communicate at full line rate.

Q6.3 (2 points) What is the maximum number of servers per rack, such that each server on the left side can send data to a corresponding server on the right side, at full line rate?

Assume that each server still has its own link to the adjacent router.

Your answer should be a single integer.

4

Solution:

Rack 1 has 4 servers, each with its own link to R1.

R1 has 4 outgoing links to R7/R8/R9/R10, one for each server.

Each of R7/R8/R9/R10 has an outgoing link to R4, one for each server.

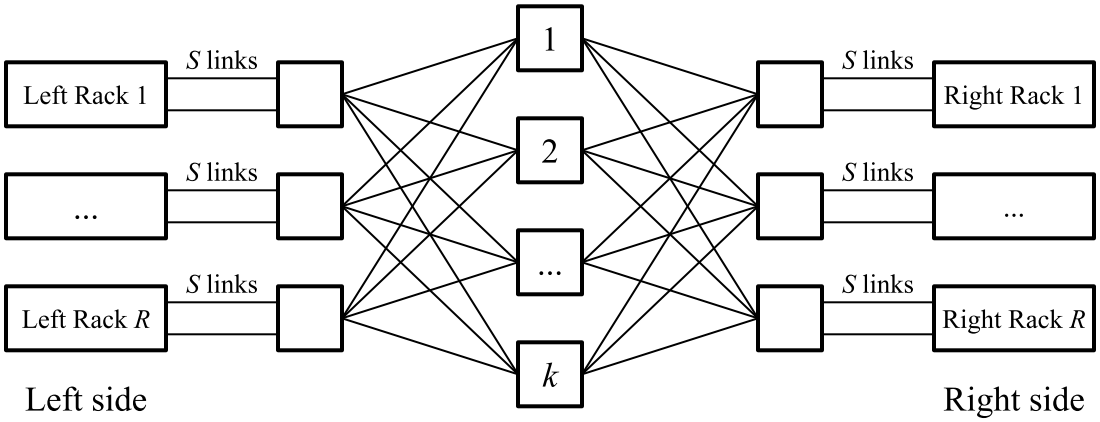
R4 has 4 links to Rack 4, one for each server.

The same pattern can be used to create dedicated paths from all servers on Racks 2 and 3, to a corresponding server on Racks 5 and 6.

(Question 6 continued...)

For the rest of the question, suppose we generalize the topology:

- There are R racks on the left side, and R racks on the right side. ($R = 3$ in the example.)
- There are S servers per rack. ($S = 2$ in the example.)
- There are k servers in the middle layer. ($k = 4$ in the example.)



Q6.4 (2 points) What values of k allow each server on the left side to send data to a corresponding server on the right side, at full line rate?

Fill in the inequality. Your answer could be in terms of R, S, and/or k.

$k \geq S$

Solution:

Consider a router at the first layer (directly connected to a left rack). This router has S incoming links, so it must have at least S outgoing links to give a dedicated link to each server.

Each outgoing link goes to a different middle-layer router, so we must have at least S middle-layer routers. In other words, $k \geq S$.

Q6.5 (3 points) How many total links are used to build this topology?

Your answer could be in terms of R, S, and/or k.

$2RS + 2Rk$

Solution:

Each rack has S links, and there are $2R$ racks in total, for a total of $2RS$ links connected to a rack.

For links interconnecting routers, each of the R routers on the left side must connect to each of the k routers in the middle, for a total of Rk links. Then, since the topology is symmetric, there are another Rk links connecting the middle routers to the right-side routers, for a total of $2Rk$ links interconnecting routers.

Reminder: The radix of a switch is the number of ports that switch has.

Assume that each switch has exactly the number of ports needed for the topology (i.e. no additional unused ports).

Q6.6 (1 point) The radix of every switch in this topology is the same.

Ⓐ True Ⓑ False

Solution:

A router in the left layer has radix $S + k$.

A router in the middle layer has radix $2R$, which is different.

Q6.7 (2 points) What value of k causes every switch in the topology to have the same radix?

Fill in the expression. Your answer could be in terms of R, S, and/or k.

$k = 2R - S$

Solution:

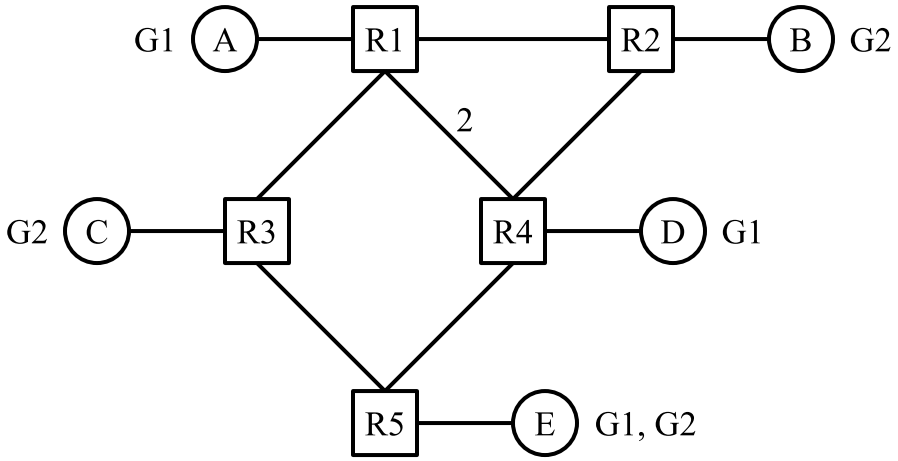
From the previous subpart, we just need to set the two radices equal to each other:

$S + k = 2R$

Simplifying gives us $k = 2R - s$.

Q7 *Multicast* **(14 points)**

Consider the topology below. All unlabeled links cost 1. Each host belongs to either group G1, or group G2, or both.



In the next four subparts, E wants to send a multicast packet to all other members of G1, using DVMRP.

Q7.1 (3 points) Before any pruning is performed, which links will be used to forward this packet? Select all that apply.

- [A] R1—R2
- [B] R1—R3
- [C] R1—R4
- [D] R2—R4
- [E] R3—R5
- [F] R4—R5

Solution:

Before any pruning is performed, the packet will be forwarded to all hosts (even the ones that are not members of G1).

Because we're using DVMRP, the packet will be forwarded along the shortest-paths tree rooted at E, and touching every other host.

This tree has the following shortest paths:

- To A, use path E—R5—R3—R1—A. Includes edges R3—R5 and R1—R3.
- To B, use path E—R5—R4—R2—B. Includes edges R4—R5 and R2—R4.
- To C, use path E—R5—R3—C. Includes edge R3—R5.
- To D, use path E—R5—R4—D. Includes edge R4—R5.

When building the E-to-G1 DVMRP tree, _____ (i) sends a prune message to _____ (ii).

Q7.2 (1 point) Blank (i): Who sends the prune message?

- (A) R1
- (B) R2
- (C) R3
- (D) R4
- (E) R5
- (F) R6

Solution: See solution to next subpart.

Q7.3 (1 point) Blank (ii): Who is the prune message sent to?

Ⓐ R1 Ⓑ R2 Ⓒ R3 Ⓓ R4 Ⓔ R5 Ⓕ R6

Solution:

R2′s only child on the tree is B, and B is not a member of G1. Therefore, R2 can send a prune message to its parent.

R2′s parent is its next-hop to E, which is R4.

Q7.4 (2 points) At convergence, is R3 part of the E-to-G1 DVMRP tree? In other words, is R3 forwarding packets from E to G1?

Ⓐ Yes, because R3 has a child who is not pruned.

Ⓑ Yes, because R3 is directly-connected to a G1 member.

Ⓒ No, because all of R3′s children have been pruned.

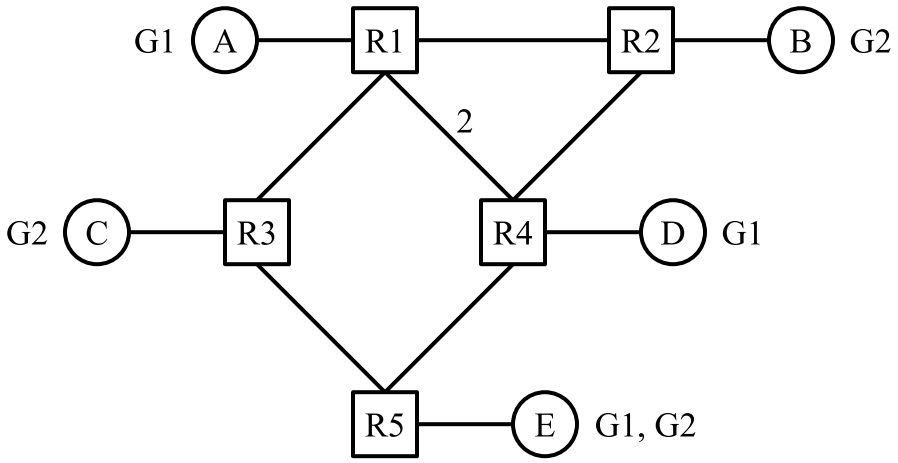
Ⓓ No, because R3 is not directly-connected to any G1 member.

Solution:

R3 is not directly connected to a G1 member. C is not a G1 member.

However, R3′s child is R1, and R1 is connected to A, which is a member of G1. Therefore, R3 will participate in forwarding packets from E to G1, by receiving packets from R5 and forwarding them to R1.

The diagram, reprinted for your convenience:



Q7.5 (2 points) In this subpart only: Suppose that some time later, B decides to join group G1.

According to the DVMRP protocol from lecture, when will B start to receive multicast packets sent to G1?

Ⓐ Immediately after B joins G1.

Ⓑ Immediately after R2 learns that B joined G1.

Ⓒ Immediately after pruning state is cleared at all the routers.

Ⓓ B will never receive multicast packets sent to G1.

Solution:

Before B joins the group, R4 has pruned R2, so multicast packets to the group will not be forwarded to R2 or B.

Even after B joins the group, and after R2 learns about this fact, R4 still has R2 pruned, so B will not receive multicast packets to this group.

In DVMRP (from lecture), pruning state is periodically cleared. After pruning state is cleared, the packet is once again forwarded to everybody (including B). This time, R2 will not send a prune message, allowing B to continue receiving multicast packets to this group.

(Note: In real life, some variants of DVMRP have graft messages for a child rejoining the tree, but this wasn't discussed in lecture.)

In the next two subparts, C wants to send a multicast packet to all other members of G2, using CBT.

Suppose R4 is chosen as the core, and the routing state has converged (i.e. all G2 members have sent Join messages to the core).

Q7.6 (3 points) Which links will be used to forward this packet? Select all that apply.

- [A] R1—R2
- [B] R1—R3
- [C] R1—R4
- [D] R2—R4
- [E] R3—R5
- [F] R4—R5

Solution:

We need to build a spanning tree rooted at R4 that touches all G2 members. This has the following shortest paths:

- To B, use R4—R2—B. Includes edge R2—R4.
- To C, use R4—R5—R3—C. Includes edges R4—R5 and R3—R5.
- To E, use R4—R5—E. Includes edge R4—R5.

Q7.7 (2 points) When R4 is the core, which statement is true about the paths the packet takes from C to all other G2 members?

- (A) The packet takes the shortest paths to all G2 members.
- (B) The packet takes the shortest paths to some, but not all, G2 members.
- (C) The packet does not take the shortest path to any G2 members.

Solution:

Since C is a member of G2, the multicast packet is forwarded along the entire tree.

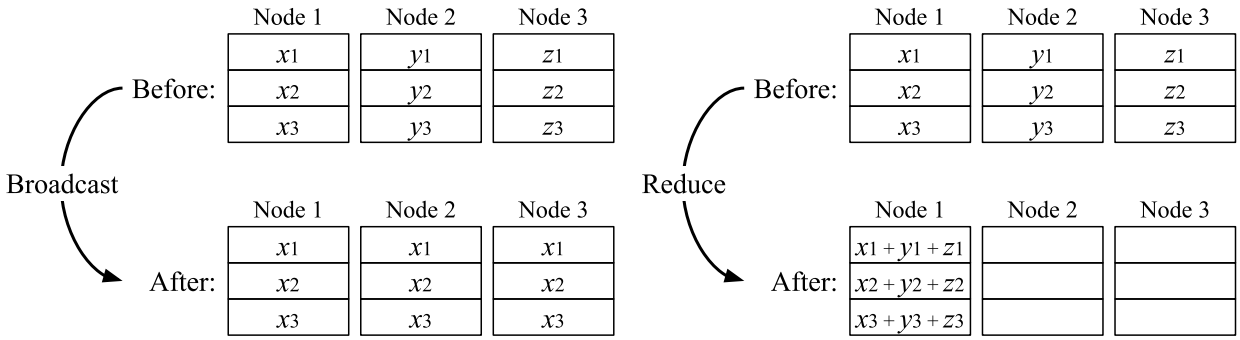
The packet is forwarded along the path C—R3—R5—E, which is the shortest path to E.

The packet is also forwarded along the path C—R3—R5—R4—R2—B. This is not the shortest path to B, because C—R3—R1—R2—B is shorter.

Therefore, the packet takes the shortest paths to some, but not all, G2 members.

Q8 *Collectives* **(8 points)**

Recall the Broadcast and Reduce collective operations from lecture, with Node 1 as the root node:



Q8.1 (3 points) Suppose we start with the "Before" state shown above. We run a Broadcast operation, immediately followed by a Reduce operation (using the Broadcast output as the input to Reduce).

What are the resulting values at Node 1?

Write one expression per box. Your expression could be in terms of: $x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$.

Node 1:

Solution: $3x_1$

Solution: $3x_2$

Solution: $3x_3$

Solution:

After the Broadcast operation, each node contains $[x_1, x_2, x_3]$.

The Reduce operation sums up the first value at each node: $x_1 + x_1 + x_1 = 3x_1$, and writes this value into the first value at Node 1.

Likewise, we sum up the second value at each node: $x_2 + x_2 + x_2 = 3x_2$, and writes this value into the second value at Node 1.

Finally, we sum up the third value at each node: $x_3 + x_3 + x_3 = 3x_3$, and writes this value into the third value at Node 1.

Q8.2 (1 point) Are the Broadcast and Reduce operations duals of each other?

Ⓐ Yes Ⓑ No

Solution:

The Broadcast operation reads from Node 1, and writes to all nodes.

The Reduce operation reads from all nodes, and writes to Node 1.

Since they read from and write to the same set of nodes, the operations are duals of each other.

In the next two subparts, we connect the 3 nodes in a ring topology, and implement Broadcast and Reduce using a similar approach as naive ring-based AllReduce. Assume each vector (e.g. $[x_1, x_2, x_3]$) is D bytes.

Your answers below can be an expression, possibly in terms of D. Give exact answers (not big-O bounds). Count all data the node sends across all time steps, but don't count data received.

Q8.3 (2 points) To implement the **Broadcast** operation on 3 nodes, what is the maximum amount of data sent by any single node?

D

Solution:

Node 1 sends its entire vector (D bytes) to Node 3.

Node 3 receives the vector and sends the entire vector (D bytes) to Node 2.

Q8.4 (2 points) To implement the **Reduce** operation on 3 nodes, what is the maximum amount of data sent by any single node?

D

Solution:

Node 1 sends its entire vector (D bytes) to Node 3.

Node 3 receives the vector $[x_1, x_2, x_3]$, and sums it with its own vector to get $[x_1 + z_1, x_2 + z_2, x_3 + z_3]$. Then, Node 3 sends this vector (D bytes) to Node 2.

Finally, Node 2 receives $[x_1 + z_1, x_2 + z_2, x_3 + z_3]$, and sums it with its own vector to get $[x_1 + y_1 + z_1, x_2 + y_2 + z_2, x_3 + y_3 + z_3]$. Then, Node 2 sends this vector (D bytes) to Node 1.

Note that the answers to this subpart and the previous subpart are the same, because Broadcast and Reduce are duals of each other, and therefore use roughly the same amount of bandwidth to complete their operations.

Comment Box

Congrats for making it to the end of the exam! Leave any thoughts, comments, feedback, or doodles here.

Nothing in the comment box will affect your grade.

Ambiguities

If you feel like there was an ambiguity on the exam, you can put it in the box below.

For ambiguities, you must qualify your answer and provide an answer for both interpretations. For example, "if the question is asking about A, then my answer is X, but if the question is asking about B, then my answer is Y." You will only receive credit if it is a genuine ambiguity and both of your answers are correct. We will only look at this box if you request a regrade.