Exam 4  
Spring 2024: Released

| Question # | Topic | Total |
| --- | --- | --- |
| 1 | Disk Size Calculation | 8 |
| 2 | Disk Scheduling | 12 |
| 3 | Networking | 15 |
| 4 | Networking Routing Algorithms | 9 |
| 5 | FAT Disk Allocation | 10 |
| 6 | Unix Commands | 7 |
| 7 | Network Transmission Time Calculation | 9 |
| Total |  | 70 |
| Hidden | —---------------------------------------------------- | 30 |

**Note:** Released questions will not change conceptually, but can change quantitatively. For example, numbers, instructions, order of events, etc. may be different in the exam. **In the actual exam, any such changes will be bolded.**

# Question 1 - Disk Size Calculation (8 points)

Suppose we have a hard disk drive with the following specifications:

* Number of platters: **15**
* Surfaces per platter: **2**
* Tracks per surface: **20000**
* Sectors per track: **400**
* Bytes per sector: **2048 bytes**
* Rotational Speed: **4000 RPM**
* Average seek time: **20 ms**

## Q1.1 (2 points)

How many bytes of data is stored on each platter of the hard disk drive? Leave your answer as a math expression: eg. 2 \* 4 \* (6 \* 9 + 5)

## Q1.2 (3 points)

How much time, in milliseconds, does it take to get to (access) a random sector on the disk? Do NOT include units in your answer. Round to 1 decimal place if needed.

Average Seek Time

Rotational Latency   
Average Rotational Latency

Time to access a Random Sector

## Q1.3 (3 points)

Now suppose we have a different hard disk drive that utilizes zoned bit recording with the following specifications:

* Number of platters: **10**
* Surfaces per platter: **2**
* Bytes per sector: **2048 bytes**
* Rotational Speed: **9000 RPM**
* Average seek time: **10 ms**
* **4 Zones:**
  + Zone 0 (Outermost zone):
    - 24 tracks, 48 sectors per track
  + Zone 1:
    - 12 tracks, 30 sectors per track
  + Zone 2:
    - 6 tracks, 20 sectors per track
  + Zone 3:
    - 1 tracks, 6 sectors per track

What is the total capacity of this new hard drive in bytes? Leave your answer as a math expression: eg. 2 \* 4 \* (6 \* 9 + 5)

# Question 2 - Disk Scheduling (12 points)

We have a disk with the following specifications:

* **Total # of cylinders:** 350 (numbered 0 to 349). The outermost cylinder = 0 and the innermost cylinder = 349.
* **Current head position** = cylinder 190, and the head is currently moving inwards (moving towards cylinder 349)

There are 10 pending requests shown below in order of arrival from left to right. Assume that each request transfers an entire track, and, therefore, the disk head can immediately start transferring the track after locating it (i.e., there is no rotational latency incurred to locate a sector).

| **Request ID** | | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cylinder ID** | | 34 | 50 | 2 | 100 | 49 | 36 | 200 | 250 | 1 | 10 |

The schedule for a scheduling algorithm is the list of cylinder IDs where the head stops to read a track and/or to change direction.

## Q2.1 (3 points)

Show the schedule for the SCAN scheduling algorithm. Write your answer as a comma-separated list of cylinder IDs: eg. 109, 231, 4, 70

200, 250, 349, 100, 50, 49, 36, 34, 10, 2, 1, 0

## Q2.2 (3 points)

Show the schedule for the C-SCAN scheduling algorithm. Write your answer as a comma-separated list of cylinder IDs: eg. 109, 231, 4, 70  
200, 250, 349, 0, 1, 2, 10, 34, 36, 49, 50, 100

Suppose that the latency for moving the head position is **M** per cylinder, and the latency of transferring a track is **C**. Give your answers in terms of M and C.

## Q2.3 (3 points)

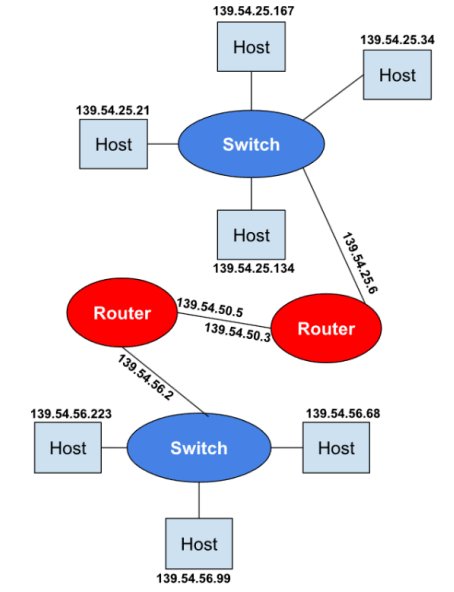
What is the expected response time for request R10 if the disk scheduling algorithm is SSTF? Answer in terms of M + C: (eg. 49M + 21C)

## Q2.4 (3 points)

What is the expected response time for request R3 if the disk scheduling algorithm is C-LOOK? Answer in terms of M + C: (eg. 49M + 21C)

# Question 3 - Networking (15 points)

Consider the following chunk of a network. All IPs in the figure are IPv4 addresses with a subnet mask of 255.255.255.0.



## Q3.1 (2 points)

How many network layer hops would it take to go from host 139.54.56.99 to host 139.54.25.34?

2 or 3

## Q3.2 (2 points)

How many link layer hops would it take to go from host 139.54.56.99 to host 139.54.25.34?  
4 or 5

Nityam is responsible for managing a real-time video streaming service that delivers live broadcasts of sports events to viewers worldwide. The service needs to ensure minimal latency and can tolerate some packet loss without significant impact on user experience.

## Q3.3 (2 points)

Which transport layer protocol should Nityam use?  
Nityam should use UDP.

## Q3.4 (6 points)

Explain what characteristic of your chosen protocol makes it preferable for this exchange. Contrast it with the one not chosen in Q3.3.   
UDP involves simply sending mass amounts of data to the recipient without regard for any acknowledgement. TCP/IP, on the other hand, sends data in packets and waits for acknowledgements. The tradeoff with UDP is that it prioritizes efficiency/speed over accuracy. This is optimal for a real-time video broadcast since massive amounts of data are transmitted for video, and it's fine if a few pixels here and there are slightly inaccurate if it means the video can be sent almost instantly.

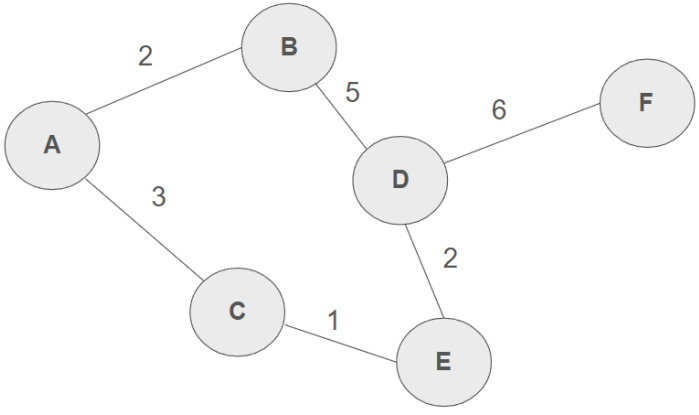
## Q3.5 (3 points)

If all the following IP addresses have a subnet mask of 20 bits, which ones are in the same subnet as the device with IP 118.211.30.56? Select all that apply.

* ~~118.211.31.23~~
* 118.211.34.56
* 118.211.255.18
* ~~118.211.16.240~~
* 118.212.30.56
* 120.211.12.56
* 118.211.1.138

# Question 4 - Networking Routing Algos (9 points)

Suppose we have the following network. At each time step starting from t=1, every node sends its distance vector (i.e., a given node’s latency to all the other nodes) to its neighbor. At t=0, all the nodes know the latencies to their respective immediate neighbors.



At time t=0, the distance vector for A is as follows:

A: 0, B: 2, C: 3, D: x, E: x, F: x

Use x to represent an unknown/infinite distance

## Q4.1 (4 points)

Compute the distance vector for A at time **t=2**. Write your answer as a comma-separated list of distances of all nodes in alphabetical order from A, including A.

Example answer for distance vector for A at t=0: 0, 2, 3, x, x, x

0, 2, 3, 6, 4, 13

## Q4.2 (5 points)

Compute a link-state routing table at node A for the network after a long period of time has passed. Assume that A already knows the whole topology of the network.

Write your answer as a comma-separated list of distances of all nodes in alphabetical order from A, including A.

Answer format: <dist to A>, <dist to B>, …, <dist to F>

Example answer for link-state routing table at A: 0, 2, 3, 4, 5, 60, 2, 3, 6, 4, 12

# Question 5 - FAT Disk Allocation (10 points)

Assume that we are using a File Allocation Table (FAT) disk allocation scheme that manages 12 data blocks numbered from 1 to 12. The table consists of 4 columns containing metadata for the FAT scheme:

1. Block number (Labeled from 1 to 12)
2. A busy bit
3. The name of the file, or NA if the block is unallocated
4. A block number for the next disk block of that file, or -1 if that block is the corresponding file’s last block.

| Block Number | Busy | File Name | Next |
| --- | --- | --- | --- |
| 1 | 1 | Karan | -1 |
| 2 | 1 | Nami | 7 |
| 3 | 1 | Fizz | 4 |
| 4 | 1 | Fizz | -1 |
| 5 | 0 | NA | -1 |
| 6 | 1 | Fizz | 12 |
| 7 | 1 | Nami | 8 |
| 8 | 1 | Nami | -1 |
| 9 | 1 | Diana | -1 |
| 10 | 0 | NA | -1 |
| 11 | 1 | Diana | 9 |
| 12 | 1 | Fizz | 3 |

## Q5.1 (6 points)

List the data blocks that comprise each file in the correct order.

Write your answer as comma-separated list of block numbers for each file: eg. 8, 1, 4, 6

Fizz: 6, 12, 3, 4

Nami: 2, 7, 8

Karan: 1

Diana: 11,9

## Q5.2 (4 points)

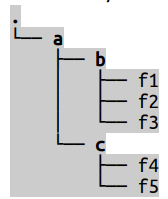
Explain one advantage and one disadvantage of the File Allocation Table (FAT) scheme  
FAT leads to efficient allocation times, especially for random access since the FAT contains the next block pointers for a given file.

Logical partitioning of a disk leads to a level of management of the space on the disk for the end user that creates artificial scarcity of space on the disk in a particular partition even

when there is plenty of physical space on the disk

# Question 6 - Unix Commands (7 points)

Given the following initial directory structure and file contents, write the command you would use to achieve each of the following outcomes.

****

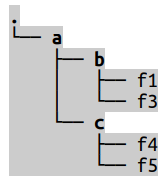
| **File Name** | **Contents** |
| --- | --- |
| f1 | foo |
| f2 | bar |
| f3 | baz |
| f4 | fred |
| f5 | waldo |

Each question below is independent of each other and should each assume restarting from the initial state.

* You are executing your commands from the parent directory of “a”
* “b” and “c” are subdirectories of “a”

For example, in order to print the contents of f4, you would type cat a/c/f4

## Q6.1 (2 points)

****

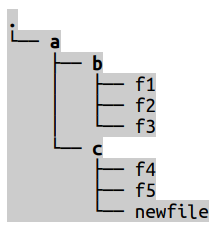
| **File Name** | **Contents** |
| --- | --- |
| f1 | foo |
| f3 | baz |
| f4 | fred |
| f5 | waldo |

What Unix command would you execute to achieve the above changes?

Write your answer as a properly formatted Unix command

rm a/b/f2

## Q6.2 (3 points)



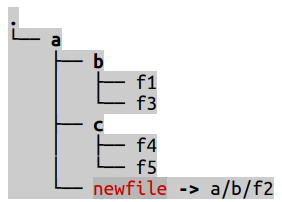
| **File Name** | **Contents** |
| --- | --- |
| f1 | foo |
| f2 | bar |
| f3 | baz |
| f4 | fred |
| f5 | waldo |
| newfile |  |

What Unix command would you execute to achieve the above changes?

Write your answer as a properly formatted Unix command

touch a/c/newfile

## Q6.3 (3 points)



| **File Name** | **Contents** |
| --- | --- |
| f1 | foo |
| f2 | bar |
| f3 | baz |
| f4 | fred |
| f5 | waldo |
| newfile |  |

After executing a command to achieve the above, you run 3 more in succession. The following is the terminal output for doing so:

$ <unknown command>

$ cat newfile

bar

$ rm a/b/f2

$ cat newfile

error: no such file or directory

What was the <unknown command> you executed to achieve the above results?

Write your answer as a properly formatted Unix command

ln -s a/b/f2 a/newfile

# Question 7 - Network Transmission (9 points)

Suppose now that you are trying to send a 30,000 byte file to your friend, and you’d like to take advantage of TCP for the transmission. Suppose also that

* The size of each TCP packet is 2000 bytes
* The size of a TCP packet header is 500 bytes
* We have a fixed window size of 5 packets
* Time of flight is 11 ms
* Assume negligible overhead at sender and receiver.
* Wire bandwidth is 18 Mbps
* Assume for simplicity that an ACK has negligible transmission delay

## Q7.1 (2 points)

What is the total number of packets you will need to send to your friend?

## Q7.2 (1 point)

How many pipelined transmission windows will you need to send over all the packets?

## Q7.3 (4 points)

Compute the end-to-end transmission time, in milliseconds, to send **1 window** of packets. Assume the window contains packets with full payloads.

Do NOT include units in your answer. Round to 1 decimal place if needed.  
  
Wire Delay for 1 packet

## Q7.4 (2 points)

Still using sliding window transmission, compute the end-to-end transmission time, in milliseconds, to send **all packets**. Assume the window contains packets with full payloads.

Do NOT include units in your answer. Round to 1 decimal place if needed.

Time until receiving 5th ack (First full window)

Time between receiving 5th & 10th ACKs   
Time until final (20th) ACK (4 full windows)