[30 points] You are asked to allocate a file according to either a File Allocation Table (FAT) or multi-level indexed allocation (UNIX inode - triply indirect). Assume that the file is 2000 disk blocks long, there are 4 KB per disk block, each pointer in the FAT occupies 4 bytes, the first index block contains 15 entries (of which 12 are direct, and one each is singly indirect, doubly indirect, and triply indirect - see slides or textbook), every other index block contains 15 entries (may be indirect depending on the nesting level), each index block entry takes 4 bytes, and unused index blocks don't count in the total memory cost, though unused entries in partially filled index blocks do count.
How many bytes are used to lay out the file when using:

a. **a FAT file system?**
   8kB.
b. **a UNIX-style file system?**
   (12*4)+(15*4)+((13*4)+(15*4*2))+((13*4)+(13*4*2)+(15*4*4)) = 8640 bytes

Now suppose that you wish to read the 1099'th block of the file. Assume each of the following counts as one search operation: moving from one element to the next in a linked list; indexing into an index block; moving from index block to the next. How many searches are needed to read block 1099 when using:

c. **the same FAT file system?**
   1099 (serial)
d. **the same UNIX-style file system?**
   4

2. [20 points] Suppose you are given a flash memory consisting of 4 KB pages, and there are 1024 pages. How many bytes of memory would the OS need to keep track of free space in the worst case (all of flash memory is free) if:

a. **a bitmap was used?**
   1024bits**.**
b. **a linked list was used? Assume 2 bytes/pointer.**
   2048 bytes.

Under what conditions would a linked list be more memory-efficient than a bitmap?
If the memory is mostly allocated and there's little free space.

3. [25 points] The read/write head of a disk is at track 97, moving towards track 199 (the highest-numbered track on the disk), and the disk request queue contains read/write requests for sectors on tracks 84, 155, 103, 96, and 197, respectively. Starting from the current head position, what is the total distance in tracks that the disk arm moves to satisfy all the pending requests for:

a. **FCFS scheduling**
   (97-84) + (155-84) + (155-103) + (103-96) + (197-96) = 244 tracks
b. **SCAN scheduling**
   (103-97) + (155-103) + (197-155) + (199-197) + (199-96) + (96-84) = 217 tracks
c. **LOOK scheduling**
   (103-97) + (155-103) + (197-155) + (197-96) + (96-84) = 213 tracks

4. [25 points] Describe the SSH protocol in detail, i.e. describe the initial handshaking phase as well as the subsequent data messaging phase. When a user supplies their login password, is it encrypted by a public key or a symmetric key, and why? Explain why or why not SSH is resilient to eavesdropping attacks, man-in-the-middle attacks, and/or replay attacks.

The handshake begins when the client initiates it by sending an encrypted N using the RSA public key of the server, which then verifies it using its private key. Client generates a key and encrypts it using the server's RSA public key and send it. Server decrypts it with its private key and uses the symmetric key provided. All following transfers are thereby encrypted.

The login password should be encrypted using the symmetric key in AES. SSH is resilient to MITM attacks since the snooper is powerless to decrypt the messages, even if they manage to intercept the data or route it through their servers, due to the sheer computational resources needed to decrypt the message without the private key. Since a new key is generated (not the RSA public/private, but the one used for AES) each time, then replay attacks are also voided.