Andrew Candelaresi
PS5

1. a. 2000 blocks * 4KBytes * 4Bytes
b.
first level index blocks = 12* 4 bytes
single indirect blocks = 15* 4 bytes
second indirect blocks = 13* 4 + 15* 4* 2 bytes
third indirect blocks = 13* 4 + 13* 4* 2 + 15* 4* 4 bytes

(12 * 4)+(15 * 4)+(13 * 4)+(15 * 4 *2)+(13 * 4)+(13 * 4 * 2)+(15 * 4 * 4) = 8640 bytes
c. 1099
d. 4th level down into the search

2. 1024 Bits = 128 bytes
1024 *2 = 2048

when there are fewer free blocks

3.
a.97,84,155,103,96,197
13+71+52+7+101 = 244

b.97, 103, 155, 197, 96, 84
6+52+42+101+12 = 213
c.97, 103, 155, 197, 84, 96
6+52+42+113+12 = 225

4 Secure Shell is a cryptographic network protocol. I provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH sever application. The connection works as follows

Client sends N (the RSA public key) to the server.
Server verifies N using p (the RSA private key)
Client generates a session key K (used as a symmetric key for AES)
Client encrypts K using N (the RSA public key) and sends the encrypted key to the server
Client uses K for AES
Server decrypts the encrypted key and uses K for AES
The SSH session is now encrypted using AES with symmetric key K. All session data sent to/ from the server is now encrypted using AES.

The login password is encrypted using the symmetric key in AES. It is resilient to eavesdropping and man in the middle attacks through host authentication since the eavesdropper can not get K with out a private key and attacker can not get any of the message because they are encrypted using K. During each SSH session a new K is generated thus making replay attacks impossible.