1. Compare Gnutella and Freenet, and tell how many advantages Freenet has over Gnuteller.

Criteria	Gnutella	Freenet
Architecture	In Gnutella, in order to find a file send request to all of the neighbors, and then those neighbors multicast the request to their neighbor until the file is found. The request has TTL in order to avoid the swamping of network with requests. No routing table is maintained at the individual machines.	In Freenet, each file is identified by a unique identifier and each machine stores a set of files, and maintains a "routing table" to route the individual requests. Searching of a file is based on the rule that "If file is found on local machine as seen from routing table, than stop else forward the request to next hop in routing table."
Anonymity	In Gnutella, there is no anonymity for the publisher, the person you are getting the file from knows who you are.	In Freenet, the anonymity of the publisher is secure.
Security	Gnutella is not resistant to external attacks. Third party can interrupt the network by malicious file deletion and denial to a particular file.	Freenet is resistant to external attacks, a third party cannot interrupt the network.
Protocol	Gnutella runs over HTTP.	Freenet uses its own port and protocol, rather than running over HTTP.
Traffic Generation	In Gnutella, if a node receives a request it cannot satisfy, it multicast request to all of its neighbors, which can result in the swamping of network with requests.	In Freenet, if a node receives a request that it cannot satisfy, it only forwards the request to only single peer, it does not multicast to all peers as Gnutella does. Thus freenet is more restrained in traffic generated than Gnutella.

Advantages of Freenet.
Almost all the differences are the advantages of Freenet over Gnutella.

2. Prove that any lookup request can be finished within $\boldsymbol{O(logN)}$ hops where N is the maximum number of nodes in Chord.

Suppose a node ***n*** receives a lookup request for the key ***k***. Let ***p*** be the immediate predecessor of ***k***. If ***n*** ≠ ***p*** then node ***n*** will forward the request to its $\boldsymbol{ith}$ successor, which is the closest predecessor of ***k*** in its finger table has. Let's call this node ***f***. Note that

$$\left(n + 2^{i-1}\right) < \boldsymbol{p}, \boldsymbol{f} < \left(n + 2^{i}\right)$$

Hence the distance between ***p*** and ***f***, $|\boldsymbol{p} - \boldsymbol{f}|$ is at most 2^{i-1}. Thus the distance between p and f is less than the distance between n and f.

$$|\boldsymbol{p} - \boldsymbol{f}| < |\boldsymbol{n} - \boldsymbol{f}|$$

And the distance between p and f is at most half of the original distance.

$$|\boldsymbol{p} - \boldsymbol{f}| < \frac{\left(n + 2^{i}\right)}{2}$$

So after ***t*** steps the distance remaining to ***p*** is at most $2^m/_{2^t}$ and after $\boldsymbol{Log\ N}$ steps the remaining distance is at most $2^m/_N$. If the distance between ***n*** and ***p*** halves in each step, and is at most $\boldsymbol{2^m}$ initially than then within ***m*** steps the distance will be one, meaning we have reached ***p***. Because nodes are distributed uniformly at random along the identifier circle, the expected number of nodes falling within an interval of this length is 1, and with high probability, there are fewer than $log\ N$ such nodes. Because the message always advances by at least one node, it takes at most $\boldsymbol{log\ \ N}$ steps for a message to traverse this remaining distance. The total expected hop count is thus $\boldsymbol{O(log\ N)}$**.**

3. Why are there some blank entries at the bottom lines of routing table? What is use of leaf set in routing process?

There are some blank entries in the routing table of pastry because the entries in the nth row of routing matrix should have n matching digits with the original node. For the smaller n it is easy to find the matching digit but as the n increases towards the bottom of the table, it is difficult to find the match for those particular cells, which results in the black entries in the routing table.

Leaf set with the size of L serves as a fall back for routing table.

4. Assume space size is (8 × 8). Node n1:(1, 2) is the first node that joins and therefore covers the entire space. n2:(2, 2) is the second node that joins. How the entire space is divided after n2 joins?

In order to maintain the uniform partitioning of the space, when node 2 joins, the whole space is split into the following zones.