x1i) Nothing is encrypted in Bitcoin, for Ethereum the peer-to-peer network connections are encrypted.

Ii) A signature is necessary to validate a transaction, or to prove that you have sufficient funds to make that transaction. (Verify the ownership of the TX; ensure the integrity of the TX that the content will not be changed; prevent double spending; not allow senders to deny the TX or non-repudiation)

B) Get hash of txn then from the hash work up the tree to find the Merkle root. Compare Merkle root with the Merkle root in the block header if it is similar then the transaction is in the block else it is not in the blockf.

C) log2 (Number of leaf nodes)

D) Alice just needs to provide the transaction as well as nodes 2, 4 and 8. Bob can then find the hash the transaction and calculate the Merkle root from other hashes provided. He can then check the Merkle root to see whether it is the same as the one of in the block header.

E) <https://en.wikipedia.org/wiki/Merkle_tree#Uses>

2i) 17476 hash/sec

Ii) 32min

III) 1024min

Iv) Digital root of x is 1+4+7+3+9+0+1+4=2+9=1+1=2, second preimage means for a given x1 it is hard to find H(x1)=H(x2) given than x1 not equal to x2. A preimage to that input is any sum of numbers that result in a digital root of 2., for example{1,2,3,4,1}. This will also given a digital root of 2.

Bi) Yes, it is valid. For execution stack look at slides

Bii) Unspent transaction outputs (UTXO) can only be used once. If two transactions use the same UTXO as input then this is a double spend. If one transaction is included in a mined block, B, then the second transaction cannot be included in any block that comes after block B (as doing so would make the second block invalid). If a block, C, is mined that includes the second transaction and C is mined on top of a block that comes before B then this would cause e fork. If enough blocks were mined on top of C then this would become the main chain and so the original transaction would no longer be considered to have happened and instead the second transaction would be the "true” transaction.

Both transactions cannot appear in the same block as transactions within a block are ordered. Whichever transaction comes second within the block would be an invalid transaction and any block containing an invalid transaction is an invalid block.