Question 1

If S' were to NOT be existentially unforgeable, there needs to either (i) exist two different messages, m and m', that signs to the same signature or (ii) two different signatures to the same message.

With (i), if S were to be existentially unforgeable, this would require H to be collision resistant such that H(m) = H(m') so that S would produce the same signature for both m and m'.

With (ii), if H were to be collision resistant, this would require S to not be existentially unforgeable so that there would be two different signatures on the same message.

Hence, if S is existentially unforgeable and H is collision-resistant, both (i) and (ii) cannot be true, thus making S' to be existentially unforgeable.

Question 2

a) Alice can recompute the Merkle root by using $T_5, T_6, H(T_1, T_2, T_3), H(T_7, T_8, T_9)$

b) $\lceil (k-1) \log_k n \rceil$

c) The proof size overhead would grow by $O(k \log_k n)$. However, an advantage of using a k-ary tree is that it reduces the computation costs of computing the inclusion proofs. This is because, with large n, a tree with a large k would be significantly shorter than a binary Merkle tree, thus requiring less information during proofs.

Question 3

a) <private key>

b) This form of a password is also not easy to memorise, so losing her phone will mean that she will most likely lose access to the private key, thus losing access to the bitcoin.

c) No it does not, While storing a hash script would require less space while providing the same security, if she stores it on her phone and loses her phone, she would still be losing access to the hash script and thus the script.

Question 4

a) Alice needs to send the transactions required in the Merkle proof to prove that her transaction is part of the block.

b) Assuming each header is approximately 80 bytes, the proof size would be $(80k)log_2 n$. With $k = 8, n = 256$, this gives us 5120 bytes.

c) If the hash value of the transaction is higher than the hash, we will not need to traverse the previous blocks. However, the worst case is still that we need to traverse all the blocks in the chain. This both leads to a best case of $O(\log_k n)$ and worst case of $O(k \log_k n)$.

Question 5

a) We can modify it so that the private key can only unlock the bitcoin wallet at the end of the week. This can be done by providing a locktime parameter on the tickets to point to the end of the week.

b) We can change the design such that the same private key cannot be used twice as the winning private key. This can be done by ensuring that we encrypt the jackpot with a new public key pair each week.