Task 1 - investigation

1. What is the point of NoSQL databases?
NoSQL Systems are horizontally scaled data storages of simple structured and unstructured data across a large cluster of nodes or hardware. Data is replicated to all or some nodes and if one of the nodes shuts down, the system continues to operate with the same data available. The data requires a fixed number of replicas and if the crashed node violates this, the data is copied over to another node that does not contain that specific data.

2. What is the CAP theorem?
- **Consistency**: Every read receives the most recent write or an error
- **Availability**: Every request receives a (non-error) response, without the guarantee that it contains the most recent write
- **Partition tolerance**: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

The CAP theorem implies that there is a shift between these three culaties, when building a distributed system. It's true that one may have to focus on 2 points more than the last, but one should never be completely neglected.

Relational databases usually weigh consistency over availability, while noSQL usually weights availability and partition tolerance over consistency.

3. What are ideal use cases of HBase?
The ideal use of HBase is in cases concerning quick access to a large amount of data and usually suited for real-time data processing.

Task 2 - Bloom filters

1. What is a bloom filter?
A Bloom filter is a data structure that focuses on space-optimization and is used to check whether an element is part of a set or not, using varius hash functions.

2. What is an advantage of bloom filters over hash tables?
The advantage of bloomfilters is the space efficiency and a seachtime better than Linear and binary search.

3. What is a disadvantage of bloom filters?

False positive - it might tell you that an element is present in the set, even though it’s not.

4. Using your language of choice, implement a bloom filter with add and check functions. The backing bit-array can simply be a long (64 bit integer).

5. If you are to store one million ASCII strings with an average size of 10 characters in a hash set, what would be the approximate space consumption?

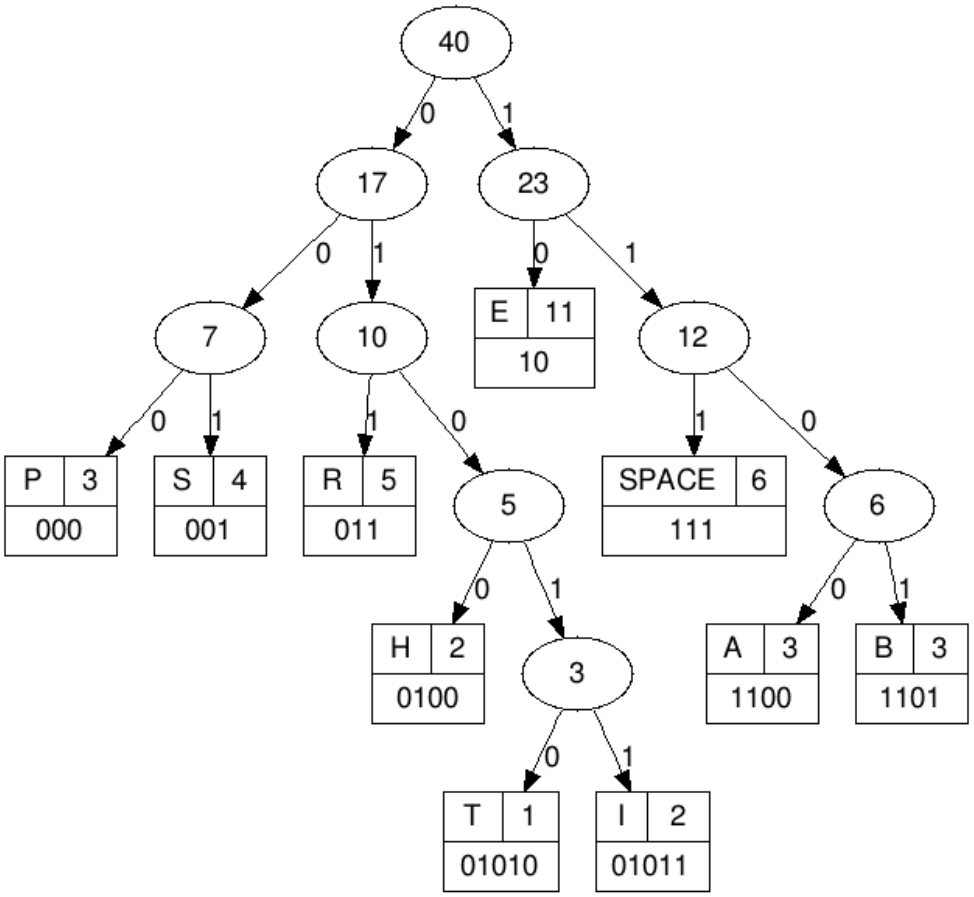
6. The following equation gives the required number of bits of space per inserted key, where E is the false positive rate.
b = 1.44log2(1/E) (1)

7. How many bits per element are required for a 1% false positive rate?

8. How many bits per element are required for a 5% false positive rate?
9. If you are to store one million ASCII strings with an average size of 10 characters in a bloom filter, what would be the approximate space consumption, given an allowed false positive rate of 5%?.

Task 3 - Huffman coding

1. Generate Humann Code (and draw the Humann Tree) based on the following string: “beebs beepps!!!!! their eerie ears hear pears"



2. How many bits is the compressed string? How many bits is the raw ASCII string?
3. Compress \pete is here" with the Hu
mann tree from before.
4. Write your own 10 word sentence. Generate the Hu
mann Code (a new
Hu
mann Tree), and write a new compressed message (ie. in binary).
Swap with one of your fellow students, and decompress each other's message.