Assignment 3 – Team Infinity

Workflow

1) Master.java
   - Add all hints of one batch to an HashMap called hintMap
   - Add every password to an ArrayList 'passwords' and create an ArrayList 'passwordChars' of the same size containing the possible chars of the passwords in each Array-List entry
   - Using the given chars of the password with length n, generate all possible sequences of length n-1 → there are n groups of sequences, in every group one of the n chars is missing
   - For every of the n different sequences we generate HintMessages with one of the sequences, the missing char and the hintMap
   - Schedule the HintMessages to all idle workers or add them to an unassignedWorkQueue

2) Worker.java
   - When Receiving a HintMessage generate all possible permutations of the given sequence with the heap algorithm, encrypt every permutation with SHA-256, and compare it to the given hints
     - As soon as one hint is cracked, we send PasswordCharMessages. It might be the case that one hint belongs to multiple password, we send a message for every password. The message contains the ID of the password and the missing char the hint refers to.
     - When all permutations are compared to the hints we send a HintsCompletedMessage to get new work for the worker

3) Master.java
   - When receiving a PasswordCharMessage the missing char of the hint is removed from the passwordChars List at the given ID
   - When receiving a HintsCompletedMessage we can assign new work to the worker
     - Next HintMessage if there is one at the unassignedWorkQueue
     - If there are no more HintMessages to be send, the password decryption can start: Every password, the remaining chars (hopefully a lot of the chars have been removed because of the hint cracking) and the length of the password are combined to a PasswordMessage
     - all PasswordMessages without a worker wait in an unassignedPasswordWork Queue

4) Worker.java
   - For all characters that might be part of the password (m characters) we use the algorithm (given in the exercise sheet) that generates all possible strings of length k that can be formed from a set of m characters (k is the given length of the password)
   - Encrypt all of the possible strings with SHA-256 and compare them to the encrypted password, as soon as there is a match send the decrypted password in a PasswordComletedMessage to the sender

5) Master.java
    - Send the received decrypted passwords to the collector (CollectMessage)
    - If there is unassigned work left, the worker sending the PasswordComletedMessage is getting more work to do
    - If all work is assigned we wait till every worker is idle. Then send the final message (PrintMessage) to the collector and terminate (unwatch workers and send poison pills)