CS2040S Recitation 5

AY 24/25 Sem 2 — github/omgeta

Q1. (a.)

Q2. (a.) Binary search, hashing values $start : mid$ locally and remotely and compare. If the hash doesn't match, recurse on the left, else recurse on the right to find the start point of corruption.

(b.) Binary search, hashing each mid on the server and check if it exists in the same position locally.

(c.) Recursive Divide-and-Conquer

(d.) Minimising data transfer by comparing hashes instead of entire files. Hashes must not collide.

(e.) No, it is a list of hashes

(f.) An arbitrary hash function may lead to hash collisions and missing photos undetected

(g.) Yes, it solves in $O(n)$.

(h.)

(i.) Invariant: preserve difference in set sizes

(j.) Ensures the hash function uniquely maps each missing photo to a value not in H_l