
Insertion in Hybrid Merkle Tree

```
void insertHybrid(string &t, vector<string> &trans, unordered_map<int, vector<string>> &mp, string &root)
{
trans.push_back(t);
int size = trans.size();
string hash = sha256(t);
int levels = ((ceil(log3(size, 3))) + 1);
mp[1].push_back(hash);
int i = 2;
while (i <= levels){
int prevSz = mp[i - 1].size();
int currSz = mp[i].size();
int neededSz = (prevSz / 3) + (prevSz % 3 == 0 ? 0 : 1);
string newVal = "";
vector<string> v = mp[i - 1];
int k = v.size() / 3 + (v.size() % 3 == 0 ? -1 : 0);
int idx = k * 3;
int cnt = 0;
for (int i = v.size() - 1; i >= idx; i--){
string s = v[i];
reverse(s.begin(), s.end());
newVal += s;
cnt++;
}
string newHash;
if (cnt == 1)
newHash = hash;
else
newHash = sha256(newVal);
if (currSz == neededSz)
mp[i][mp[i].size() - 1] = newHash;
else
mp[i].push_back(newHash);
i++;
}
if (mp[levels].size() == 1)
root = mp[levels][0];
else{
string h1 = mp[levels][0];
string h2 = mp[levels][1];
reverse(h1.begin(), h1.end());
reverse(h2.begin(), h2.end());
string newHash = sha256(h2 + h1);
root = newHash;
mp[levels + 1].push_back(root);
}
return;
}
```
