
Verification in Sparse Merkle Tree

```
bool verifySMT(string t, int pos, vector<string> &trans, unordered_map<int, vector<string>> mp, string &root){
string hash = sha256(t);
int size = trans.size();
int levels = ceil(log2(size)) + 1;
int i = 2;
while (i <= levels){
int prevSz = mp[i - 1].size();
int prevPos = pos;
if (prevPos % 2 == 0){
string h1 = mp[i - 1][prevPos - 2];
string h2 = hash;
reverse(h1.begin(), h1.end());
reverse(h2.begin(), h2.end());
hash = sha256(h2 + h1);
pos = prevPos / 2;
}
else{
if (prevPos == prevSz)
pos = (prevPos / 2) + 1;
else{
string h1 = hash;
string h2 = mp[i - 1][prevPos];
reverse(h1.begin(), h1.end());
reverse(h2.begin(), h2.end());
hash = sha256(h2 + h1);
pos = (prevPos + 1) / 2;
}}
i++;
}
if (hash == root)
return true;
return false;
}
```
