
Smart Contract of Adding Transaction Sparse Merkle Tree

```
function addTransaction(string memory trans) public {
bytes32 hash = sha256(abi.encodePacked(trans));
transactions[count]=trans;
count++;

uint256 size = count;
uint256 x1=1;
uint256 t1=0;
while(x1<size){
x1=x1*2;
t1++;
}
t1++;
uint256 levels=t1;
hashes[1].push(hash);
uint256 i = 2;
while (i <= levels) {
uint256 prevSz = hashes[i - 1].length;
if(prevSz%2==1){
uint256 currSz = hashes[i].length;
uint256 neededSz = (prevSz / 2) + 1;
if (currSz == neededSz)
hashes[i][hashes[i].length - 1] = hashes[i - 1][prevSz - 1];
else
hashes[i].push(hashes[i - 1][prevSz - 1]);
}
else{
bytes32 h1=hashes[i-1][prevSz-2];
bytes32 h2=hashes[i-1][prevSz-1];
bytes32 newHash = sha256(abi.encodePacked(h1, h2));
if(hashes[i].length>0)
hashes[i][hashes[i].length-1]=newHash;
else
hashes[i].push(newHash);
}
i++;
}

if (hashes[levels].length == 1)
root = hashes[levels][0];
else
{
bytes32 h1 = hashes[levels][0];
bytes32 h2 = hashes[levels][1];
bytes32 newHash = sha256(abi.encodePacked(h1, h2));
root = newHash;
hashes[levels + 1].push(root);
}
return;
}
```
