
Smart Contract of Verifying Transaction Sparse Merkle Tree

```
function verifyTransaction(uint position, string memory trans) public view returns (bool) {
bytes32 hash = sha256(abi.encodePacked(trans));
uint256 size = count;

uint256 x1=1;
uint256 t1=0;
while(x1<size){
x1=x1*2;
t1++;
}
t1++;
uint256 levels=t1;

uint i = 2;
while (i <= levels) {
uint256 prevSz = hashes[i - 1].length;
uint256 prevPos = position;
if (prevPos % 2 == 0){
bytes32 h1 = hashes[i - 1][prevPos - 2];
bytes32 h2 = hash;
hash = sha256(abi.encodePacked(h1, h2));
position = prevPos / 2;
}
else
{
if (prevPos == prevSz)
position = (prevPos / 2) + 1;
else
{
bytes32 h1 = hash;
bytes32 h2 = hashes[i - 1][prevPos];
hash = sha256(abi.encodePacked(h1, h2));
position = (prevPos + 1) / 2;
}}
i++;
}
if (hash == root)
return true;
return false;
}}
```
