```
void insertSMT(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(log2(size)) + 1;
   mp[1].push_back(hash);
   int i = 2;
   while (i <= levels)
   {
      int prevSz = mp[i - 1].size();
      if (prevSz % 2 == 1)
      {
         int currSz = mp[i].size();
         int neededSz = (prevSz / 2) + 1;
         if (currSz == neededSz)
         {
            mp[i][mp[i].size() - 1] = mp[i - 1][prevSz - 1];
         }
         else
         {
            mp[i].push_back(mp[i - 1][prevSz - 1]);
         }
      }
      else
      {
         string h1 = mp[i - 1][prevSz - 2];
         string h2 = mp[i - 1][prevSz - 1];
         reverse(h1.begin(), h1.end());
         reverse(h2.begin(), h2.end());
         string newHash = sha256(h2 + h1);
         if (mp[i].size() > 0)
            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;
}
```