```
vector < vector < int > > ConstructTree (
        int n , vector < vector < int > > edges ) {
    vector < vector < int > > adjl ;
    for ( int i = 0 ; i < n ; i ++ ) {
        adjl . push_back ( vector < int > ( ) ) ;
    }
    for ( auto e : edges ) {
        int u = e [ 0 ] ;
        int v = e [ 1 ] ;
        adjl [ u ] . push_back ( v ) ;
        adjl [ v ] . push_back ( u ) ;
    }
    return adjl ; }
```

```
function withdrawBalance (
        start , withdrawals ) {
    let end = withdrawals . reduce (
        ( balance , nextWithdrawal ) => {
            return nextWithdrawal <= balance ?
            balance - nextWithdrawal :
            balance ;
        }
        , start ) ;
    return end ; }
```