1 Psuedocode to identify the basic blocks

Algorithm 1 Parse JavaScript Code to Generate the AST

```
Input: JavaScript source code
Output: Abstract Syntax Tree $AST$
$AST \leftarrow$ esprima.parse(sourceCode)
return $AST$
```

Algorithm 2 Traverse the AST to Identify Basic Blocks

```
Input: Abstract Syntax Tree $AST$
Output: Basic Blocks $basicBlocks$
$basicBlocks \leftarrow [\,]$
for all node in $AST$ do
    if node.type is in [‘ExpressionStatement’, ‘VariableDeclaration’] then
        $basicBlock \leftarrow$ newBasicBlock($node$)
        $basicBlocks.append(basicBlock)$
    else if node.type in [‘IfStatement’, ‘ForStatement’, ‘WhileStatement’] then
        $basicBlocks.append($finalizeCurrentBlock($node$)$)$
    end if
end for
return $basicBlocks$
```