

BNF #1 Assignment:

Learning Abstract

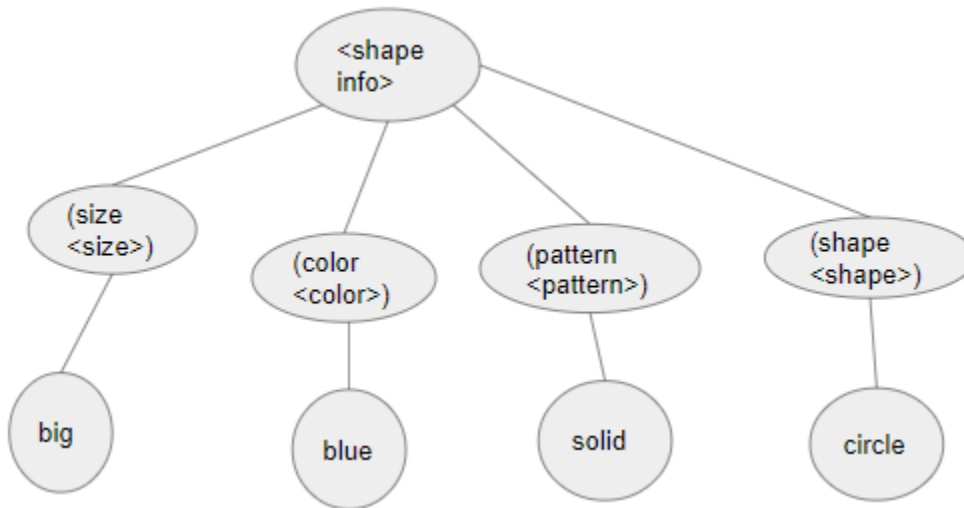
In this assignment, we take a look at the idea of BNFs. In each one of the tasks, except for the last one, each one of them looks into a different structure and how the rules of BNF dictate the concept of the problem. The last task is a summary of the concept of BNFs.

Problem 1

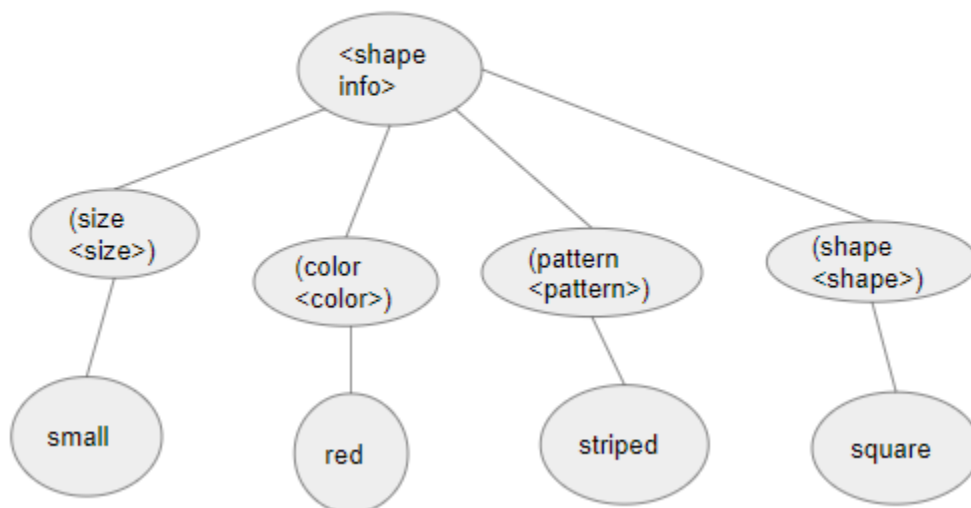
Code:

```
<shape info> ::= ( ( size <size> ) ( color <color> ) ( pattern <pattern> ) ( shape <shape> ) )  
<size> ::= big | medium | small  
<color> ::= red | yellow | blue  
<pattern> ::= striped | dotted | solid
```

Task 1:



Task 2:



Problem 2

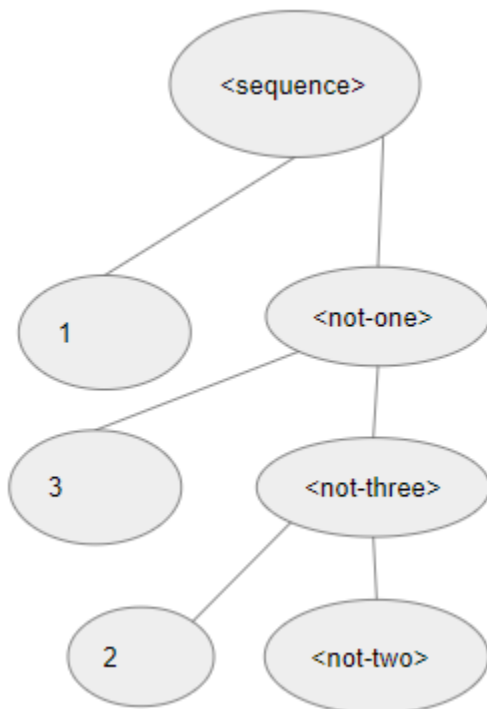
Code:

```
<sequence> ::= 0 | <not-zero>  
<not-zero> ::= 1 <not-one> | 2 <not-two> | 3 <not-three> | <empty>  
<not-one> ::= 0 <not-zero> | 2 <not-two> | 3 <not-three> | <empty>  
<not-two> ::= 0 <not-zero> | 1 <not-one> | 3 <not-three> | <empty>  
<not-three> ::= 0 <not-zero> | 1 <not-one> | 2 <not-two> | <empty>
```

Task 1:



Task 2:



Task 3:

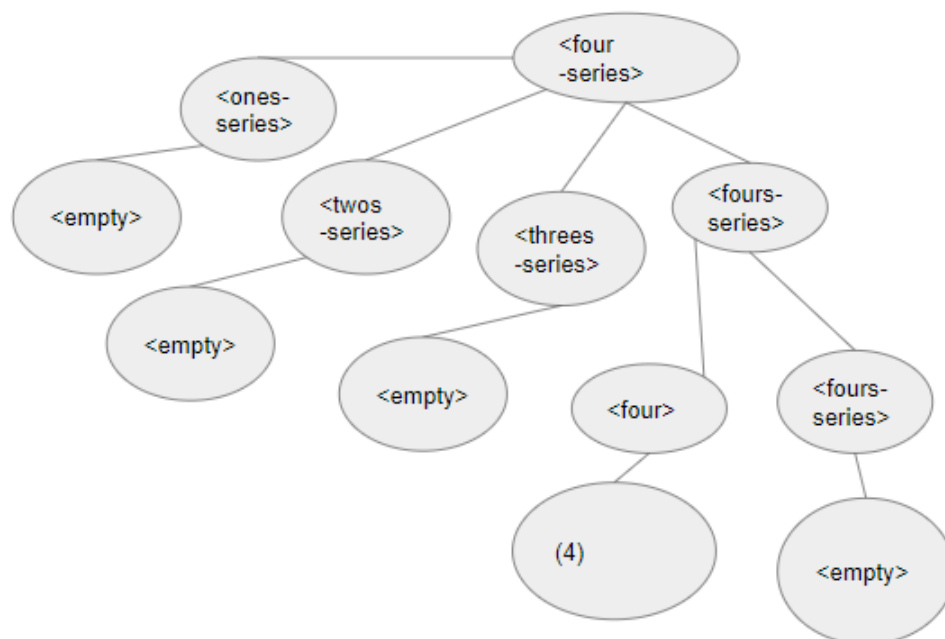
There is a consecutive pair of 2s that would cause the bnf to terminate prematurely

Problem 3

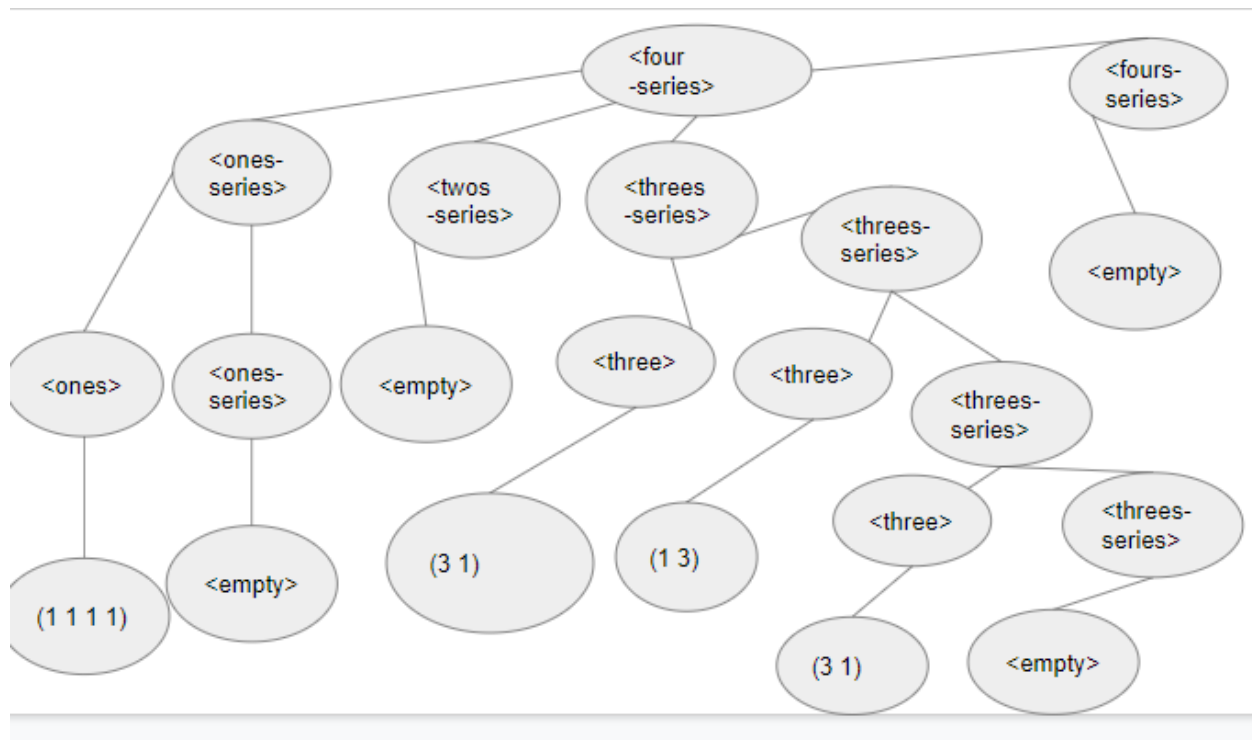
Code:

```
<four-series> ::= <ones-series> <twos-series> <three-series> <four-series>
<ones-series> ::= <empty> | <ones> <ones-series>
<twos-series> ::= <empty> | <twos> <twos-series>
<threes-series> ::= <empty> | <threes> <threes-series>
<fours-series> ::= <empty> | <fours> <fours-series>
<ones> ::= (1 1 1 1)
<twos> ::= (1 1 2) | (1 2 1) | (2 1 1)
<threes> ::= (1 3) | (3 1)
<fours> ::= (4)
```

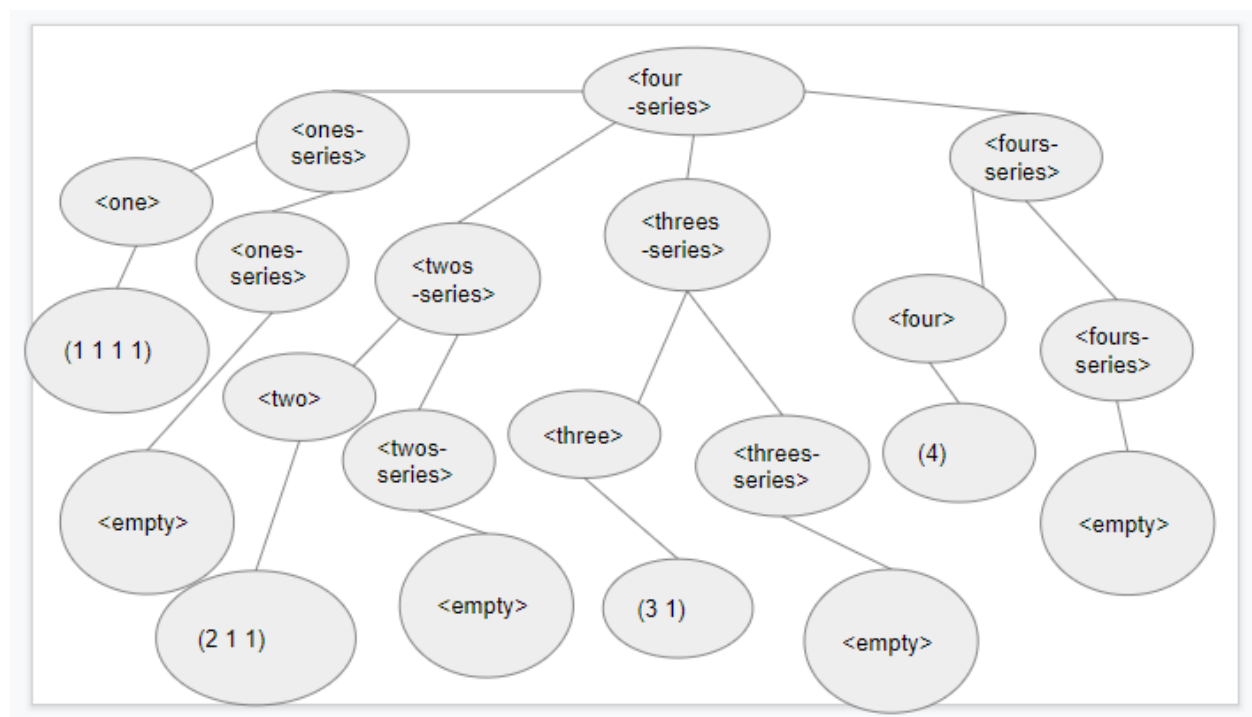
Task 1:



Task 2:



Task 3:

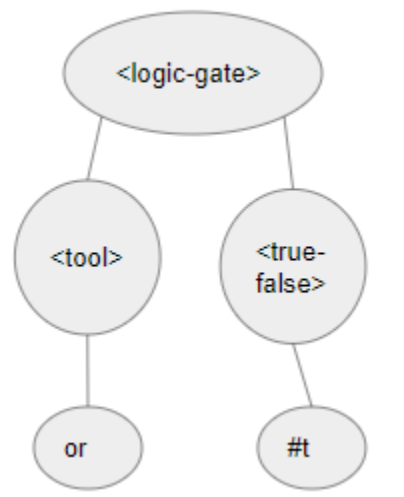


Task 4

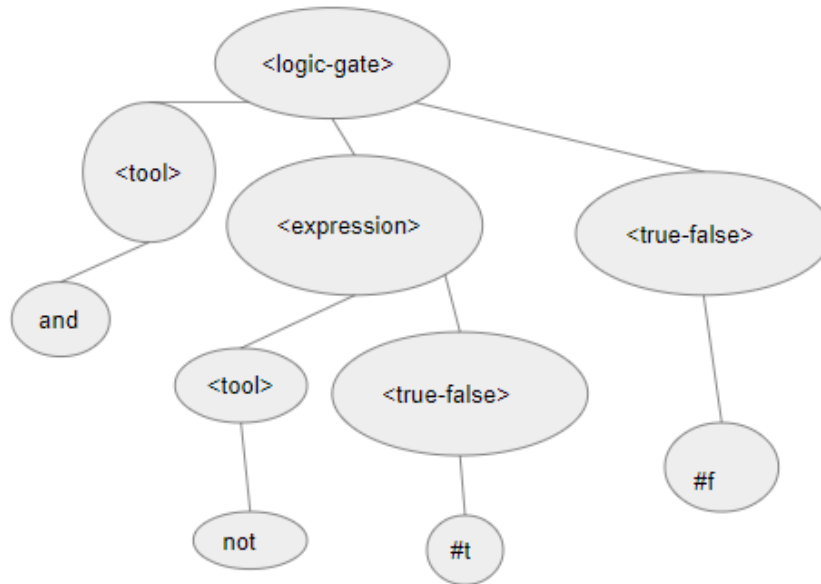
Code:

```
<logic-gate> ::= <empty> | <tool> <expression> | <tool> <true-false> | <tool> <expression>  
<expression> | <tool> <true-false> <expression> | or <true-false> <true-false> | and  
<true-false> <true-false>  
<tool> ::= not | or | and  
<expression> ::= <empty> | <tool> <true-false> | <tool> <expression>  
<true-false> ::= #t | #f
```

Task 1:



Task 2:



Task 5

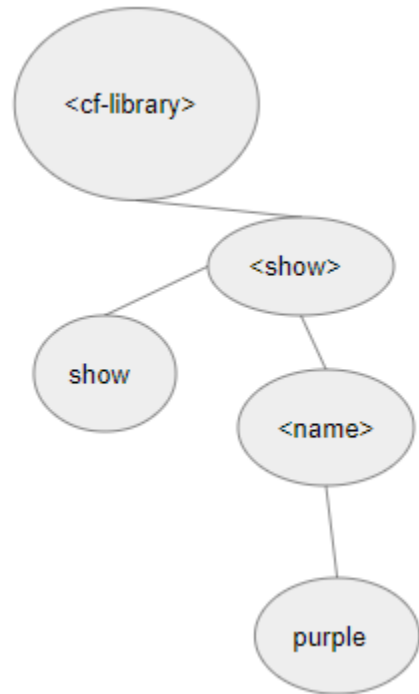
Code:

```
<cf-library> ::= <add> | colors | <describe> | <show> | exit  
<add> ::= add <rgb-numbers> <name> | add color <name>  
<rgb-numbers> ::= ( <rgb-number> <rgb-number> <rgb-number> ) | ( <rgb-number>  
<rgb-number> <rgb-number> <rgb-number> )  
<rgb-number> ::= 0 | 1 | 2 | ... | 255  
<describe> ::= describe <name>  
<show> ::= show <name>
```

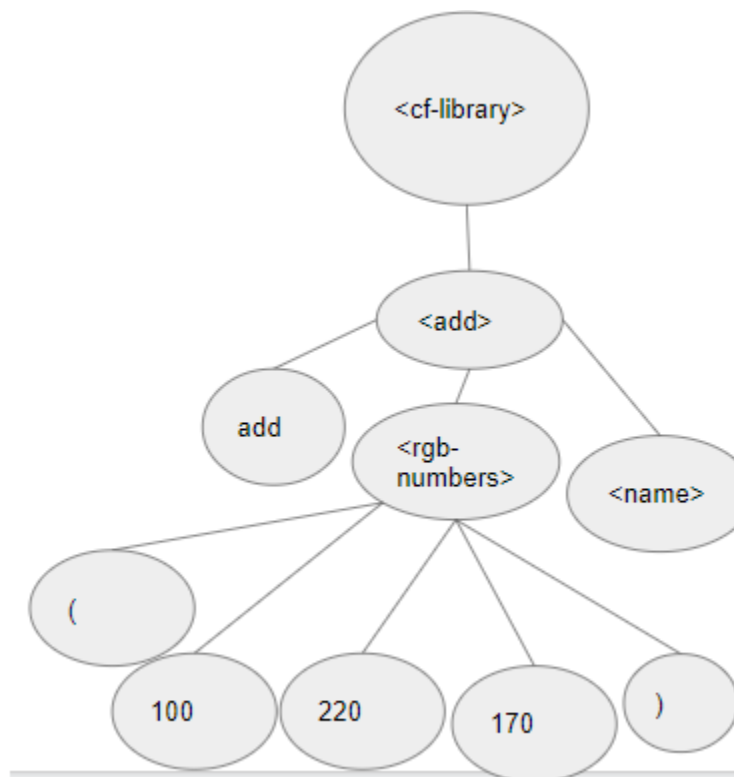
Task 1:



Task 2:



Task 3:



Problem 6

BNF is a more simplistic way of being able to capture the idea of setting these rules and boundaries to make sequences and expressions. It is like the baseline that helps push forward on how to structure and organize material.