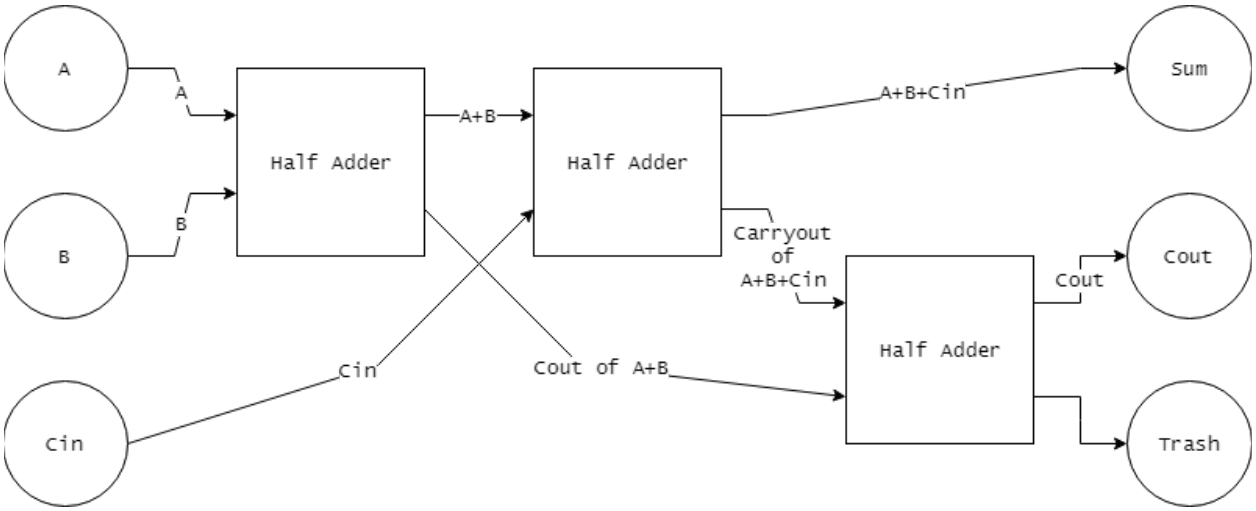
362 Homework 3

Problem 1:

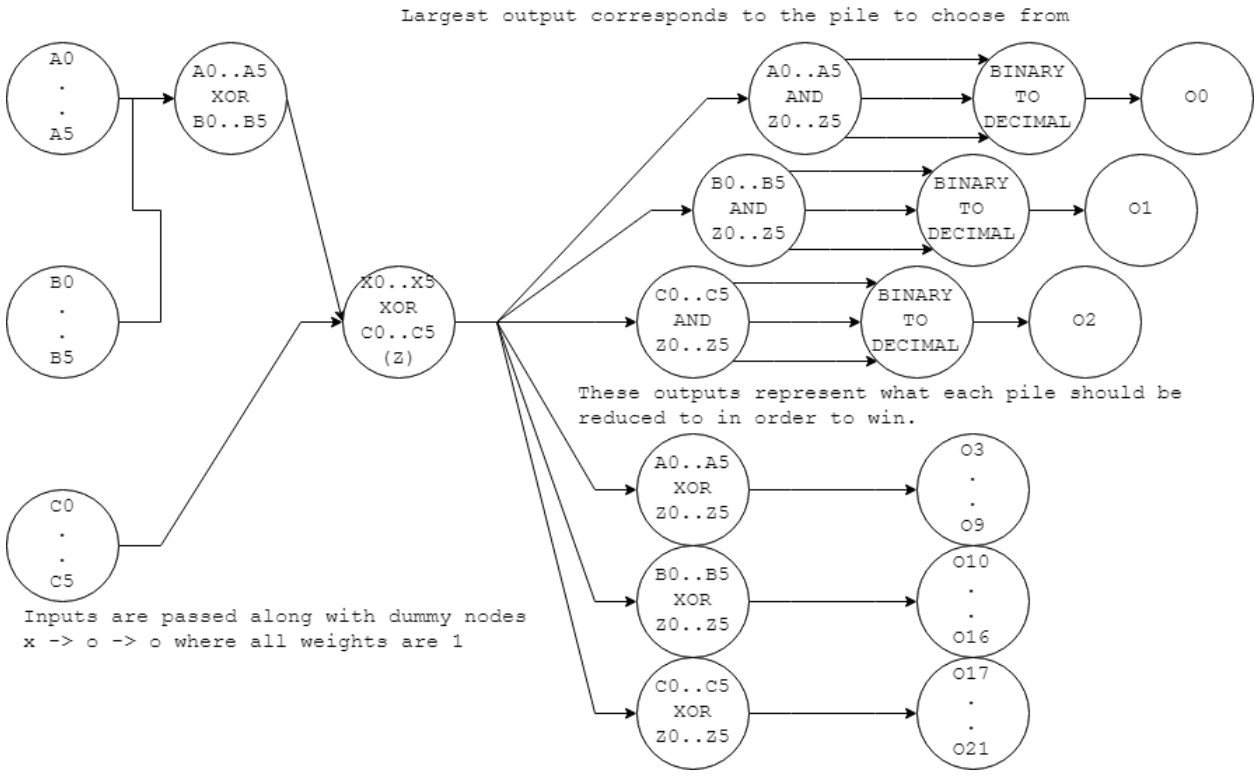


This circuit is composed of only 3 half adders. A full adder must take it an A, a B, and a carry in while a half adder only takes in an A and a B. The first (leftmost) half adds A and B. The second (middle) half adder adds the carry in to the sum of A and B. The last (rightmost) half adder adds the carry out of A+B and the carry out of A+B+Cin the sum of which is the carry out of the entire operation. All three of these half adders play a critical role and cannot be removed because of this, thus three is the smallest number of half adders that will complete the problem. The task could be accomplished with two half adders if gates other than half adders were allowed by replacing the rightmost half adder with an or gate.

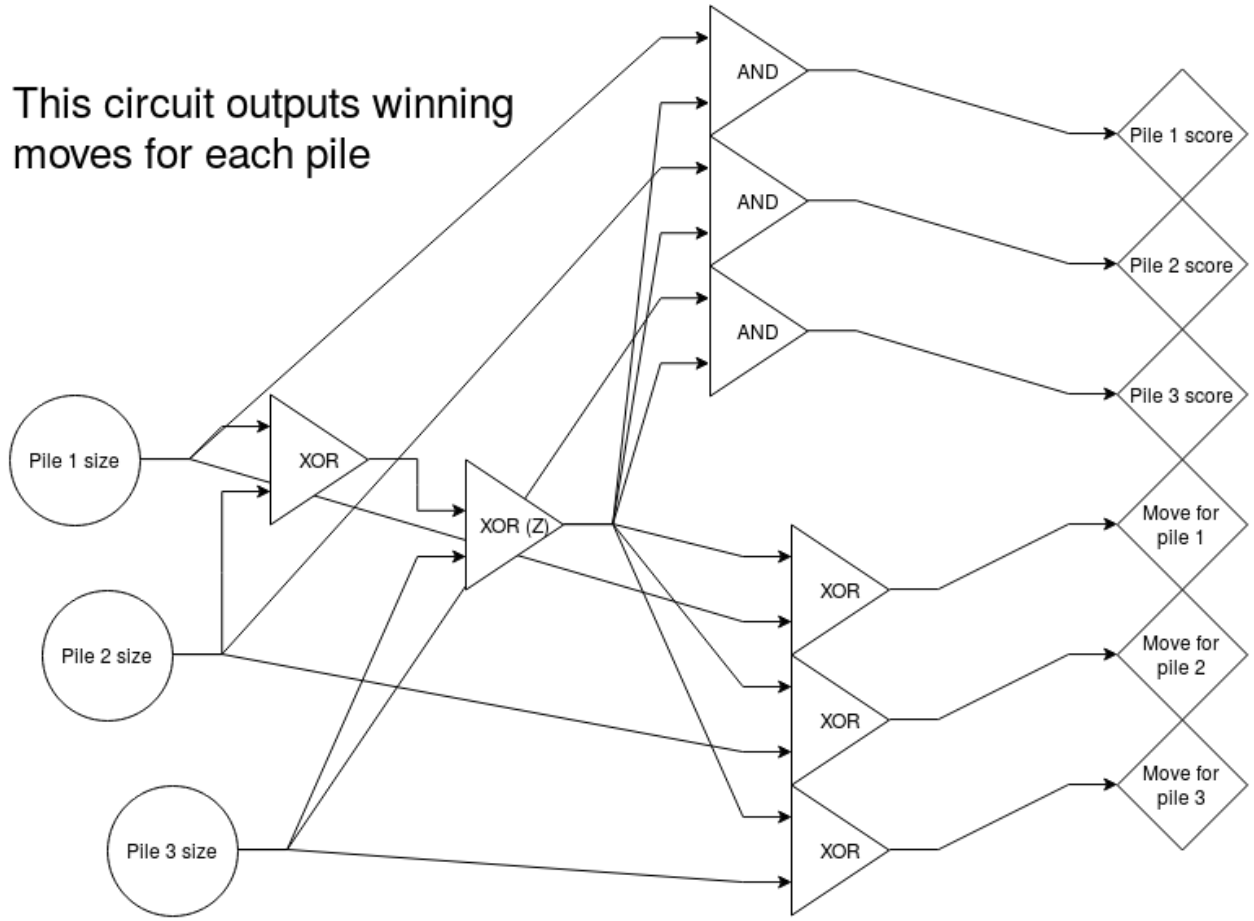
Problem 2:

This design diagram is that of my neural network for nim. To keep it clean I used black boxes that are included below in addition to the circuit version of the neural net. Unfortunately I was not able to design and implement the selection component of the problem so instead of outputting the six bits that represent what the chosen pile should be reduced to my program outputs winning pile reductions for each pile. Some of these reductions are actually illegal which is why the first 3 bits of the output must be used to pick the correct reduction.

Compile the code with java and supply 3 ints to play the game. The ints must be in the range(0,63) as per the assignment

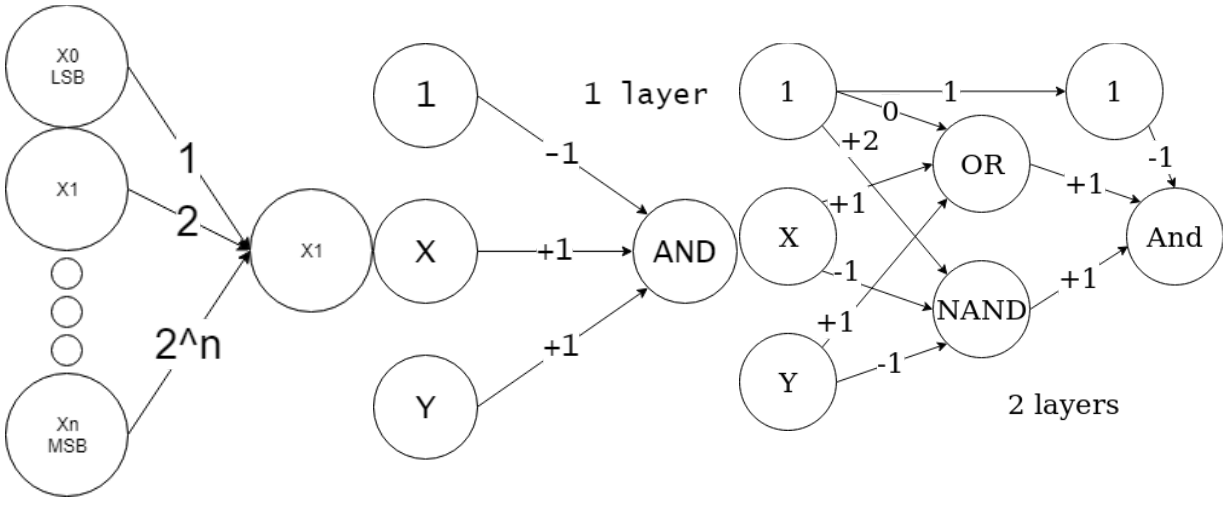

Largest output corresponds to the pile to choose from
A0 . . A5
B0 . . B5
C0 . . C5
A0..A5 XOR B0..B5
X0..X5 XOR C0..C5 (Z)
A0..A5 AND Z0..Z5
B0..B5 AND Z0..Z5
C0..C5 AND Z0..Z5
BINARY TO DECIMAL
BINARY TO DECIMAL
BINARY TO DECIMAL
O0
O1
O2
These outputs represent what each pile should be reduced to in order to win.
A0..A5 XOR Z0..Z5
B0..B5 XOR Z0..Z5
C0..C5 XOR Z0..Z5
O3 . . O9
O10 . . O16
O17 . . O21
Inputs are passed along with dummy nodes
x -> o -> o where all weights are 1

As a circuit:



Legend:
Circle: input
Square: collection of gates
Triangle: collection of gates
Diamond: output

Neuron black boxes left to right: binary to decimal, and, xor with or and nand



Game play:

With starting pile sizes 1 2 0

```
Current pile sizes: 1 2 0
Input pile number 1-3
2
Input a legal move
1
Pile 1 reduced to: 0
Current pile sizes: 0 1 0
Input pile number 1-3
2
Input a legal move
1
Human wins
```

With starting pile sizes 5 13 2

```
Current pile sizes: 5 13 3
Input pile number 1-3
3
Input a legal move
1
Pile 2 reduced to: 7
Current pile sizes: 5 7 2
Input pile number 1-3
2
Input a legal move
3
Pile 3 reduced to: 1
Current pile sizes: 5 4 1
Input pile number 1-3
1
Input a legal move
5
Pile 2 reduced to: 1
Current pile sizes: 0 1 1
Input pile number 1-3
2
Input a legal move
1
Pile 3 reduced to: 0
Neural net wins
```