EE341 Fall 2019 HW 2

Lewis Collum

Updated: September 30, 2019

github.com/LewisCollum/microelectronics

PROBLEM 4.35

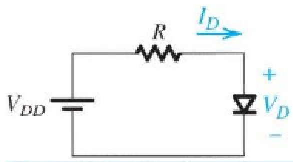


Fig. 10

4.35 Use the iterative-analysis procedure to determine the diode current and voltage in the circuit of Fig. 4.10 for $V_{DD} = 1$ V, $R = 1$ kΩ, and a diode having $I_S = 10^{-15}$ A.

Given

```
import numpy
import pint
unit = pint.UnitRegistry()

R = 1 * unit.kohm
v = {'DD': 1 * unit.V, 'D': []}
i = {'S': 10e-15 * unit.A, 'D': []}
```

Assume

$V_T = 25\text{mV}$ (thermal voltage at room temperature).

$V_{D[0]} = 0.7\text{V}$

```
v['T'] = 25 * unit.mV
v['D'].append(0.7 * unit.V)
```

Solve for $I_{D[0]}$ from diode characteristic equation

$$I_{D[0]} = I_S \cdot e^{V_{D[0]}/V_T}$$

```
i['D'].append((i['S']*numpy.exp(v['D'][0] / v['T'])).to('mA'))
print("\\noindent\\[I_{D[0]} =", f"{i['D'][0]:.3~Lx}\\]")
```

$$I_{D[0]} = 14.5\,\text{mA}$$

Solve for $I_{D[1]}$ by KVL

$$V_{DD} = I_D R + V_D$$

$$\text{So, } I_D = \frac{V_{DD} - V_D}{R}$$

```
i['D'].append(((v['DD'] - v['D'][0])/R).to('mA'))
print("\\noindent\\[I_{D[1]} =", f"{i['D'][1]:.3~Lx}\\]")
```

$$I_{D[1]} = 0.3\,\text{mA}$$

Iterative solution for V_D and I_D

$$V_{D[n]} - V_{D[n-1]} = V_T \ln \frac{I_{D[n]}}{I_{D[n-1]}}$$

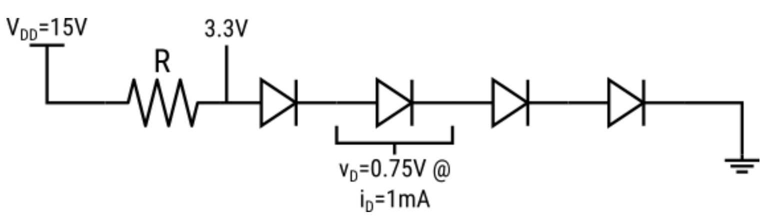
```
iterations = 7

table = [["Iteration", "$V_D$ (V)", "$I_D$ (mA)"],
         [0, f"{v['D'][0].magnitude:.6}", f"{i['D'][0].magnitude:.6}"]]

for n in range(1, iterations+1):
    v['D'].append(v['D'][n-1] + v['T']*numpy.log(i['D'][n]/i['D'][n-1]))
    table.append([n, f"{v['D'][n].magnitude:.6}", f"{i['D'][n].magnitude:.6}"])
    i['D'].append(((v['DD'] - v['D'][n])/R).to('mA'))

print(table)
```

Iteration	V_D (V)	I_D (mA)
0	0.7	14.4626
1	0.603112	0.3
2	0.610108	0.396888
3	0.609664	0.389892
4	0.609692	0.390336
5	0.60969	0.390308
6	0.609691	0.39031
7	0.609691	0.390309

PROBLEM 4.37

D 4.37 Assuming the availability of diodes for which $v_D = 0.75$ V at $i_D = 1$ mA, design a circuit that utilizes four diodes connected in series, in series with a resistor R connected to a 15-V power supply. The voltage across the string of diodes is to be 3.3 V.

Circuit Design



Given

```
import api.homework_2 as api
import pint
import math
unit = pint.UnitRegistry()

diodeCount = 4
i = {'D': [1 * unit.mA]}
v = {'DD': 15 * unit.V, 'D': [0.75 * unit.V], 'T': 25 * unit.mV, 'O': 3.3 * unit.V}
```

Solve for I_S

Finding the saturation current using the characteristic diode equation allows us to use the equation again to solve for the current of the circuit with a different voltage across the diodes.

```
i['S'] = i['D'][0]*math.exp(-v['D'][0]/v['T'])
api.printEquation("I_S", i['S'], 4)
```

$$I_S = 9.358 \times 10^{-14}\,\mathrm{mA}$$

Solve for $V_{D[1]}$ and $I_{D[1]}$

The output voltage, V_O, must be split among the diodes in series. Once we have the voltage drop for each diode, we find the corresponding current using the characteristic diode equation.

```
v['D'].append(v['O']/diodeCount)
i['D'].append(i['S']*math.exp(v['D'][1]/v['T']))
api.printEquation("V_{D[1]}", v['D'][1], 4)
api.printEquation("I_{D[1]}", i['D'][1], 4)
```

$$V_{D[1]} = 0.825\,\mathrm{V}$$

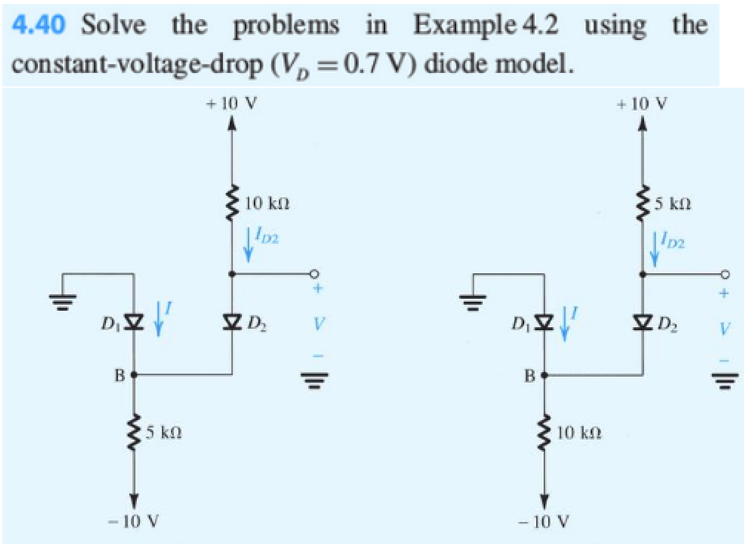
$$I_{D[1]} = 20.09\,\mathrm{mA}$$

Solve for R

```
R = ((v['DD'] - v['O'])/i['D'][1]).to('ohm')
api.printBoxedEquation("R", R, 4)
```

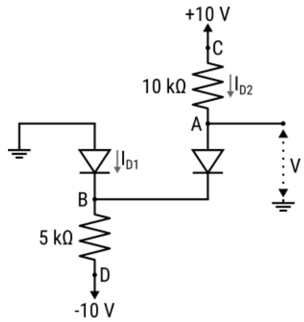
$$\boxed{R = 582.5\,\Omega}$$

PROBLEM 4.40



(A)

Circuit Label Reference



Solve for I_{D1}, I_{D2}, V_A, V_B

```
import api.homework_2 as api
from api.homework_2 import Node, Branch
unit = api.unit

node = {
    'ground': Node(0*unit.V),
    'C': Node.fromVoltage(10*unit.V),
    'D': Node.fromVoltage(-10*unit.V),
}

branch = {
    'D1': Branch.fromVoltage(0.7*unit.V),
    'D2': Branch.fromVoltage(0.7*unit.V),
}

node['B'] = Node.fromNodeBranch(node['ground'], branch['D1'])
node['A'] = Node.toBranchNode(branch['D2'], node['B'])

branch['BD'] = Branch.fromNodeToNode(node['B'], node['D'])
branch['CA'] = Branch.fromNodeToNode(node['C'], node['A'])
branch['CB'] = branch['CA'].swallowBranch(branch['D2'])

branch['BD'].setCurrentFromResistance(5*unit.kohm)
branch['CB'].setCurrentFromResistance(10*unit.kohm)
branch['D1'].setCurrentTowardsBranches(-branch['CB'], branch['BD'])

api.printBoxedEquation("I_{D1}", branch['D1'].current.to('mA'), 3)
api.printBoxedEquation("I_{D2}", branch['CB'].current.to('mA'), 3)
api.printBoxedEquation("V_A", node['A'].voltage, 3)
api.printBoxedEquation("V_B", node['B'].voltage, 3)
```

$$\boxed{I_{D1} = 0.79\,\mathrm{mA}}$$

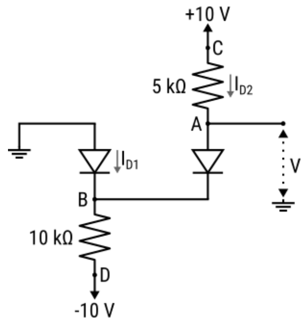
$$\boxed{I_{D2} = 1.07\,\mathrm{mA}}$$

$$\boxed{V_A = 0.0\,\mathrm{V}}$$

$$\boxed{V_B = -0.7\,\mathrm{V}}$$

(B)

Circuit Label Reference



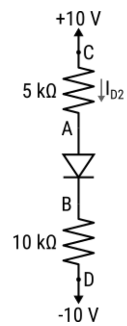
Assume both diodes are conducting

```
import api.homework_2 as api
from api.homework_2 import Node, Branch, unit

node = {'C': Node.fromVoltage(10*unit.V), 'D': Node.fromVoltage(-10*unit.V)}
branch = {'D1': Branch.fromVoltage(0.7*unit.V), 'D2': Branch.fromVoltage(0.7*unit.V)}
node['B'] = Node.fromNodeBranch(Node(0.0*unit.V), branch['D1'])
node['A'] = Node.toBranchNode(branch['D2'], node['B'])

branch['BD'] = Branch.fromNodeToNode(node['B'], node['D'])
branch['CA'] = Branch.fromNodeToNode(node['C'], node['A'])
branch['CB'] = branch['CA'].swallowBranch(branch['D2'])

branch['CB'].setCurrentFromResistance(5*unit.kohm)
branch['BD'].setCurrentFromResistance(10*unit.kohm)
branch['D1'].setCurrentTowardsBranches(branch['BD'], -branch['CB'])

api.printEquation("I_{D1}", branch['D1'].current.to('mA'), 3)
```

$$I_{D1} = -1.21\,\mathrm{mA}$$

D1 is not conducting when reverse-biased, so the assumption that both diodes are conducting is wrong.

Assume D1 is not conducting, but D2 is



Solve for V_A

The relationship between nodes A and B is

$$V_B = V_A - 0.7\mathrm{V}$$

The current through the branch can be related as

$$\frac{10\mathrm{V} - V_A}{5\mathrm{k}\Omega} = I_{D2} = \frac{V_B + 10\mathrm{V}}{10\mathrm{k}\Omega}$$

Substituting V_B for $V_A - 0.7\text{V}$ gives

$$\frac{10\text{V} - V_A}{5\text{k}\Omega} = \frac{V_A + 9.3\text{V}}{10\text{k}\Omega}$$

$$\implies 20 - 2V_A = V_A + 9.3$$

$$\implies 3V_A = 10.7$$

$$\therefore \boxed{V_A = 3.57\text{V}}$$

Solve for V_B

$$V_B = 3.57\text{V} - 0.7\text{V}$$

$$\boxed{V_B = 2.87\text{V}}$$

Solve for I_{D2}

$$I_{D2} = \frac{10\text{V} - 3.57\text{V}}{5\text{k}\Omega}$$

$$\boxed{I_{D2} = 1.286\text{mA}}$$

PROBLEM 4.46

4.46 The small-signal model is said to be valid for voltage variations of about 5 mV. To what percentage current change does this correspond? (Consider both positive and negative signals.) What is the maximum allowable voltage signal (positive or negative) if the current change is to be limited to 10%?

(A)

$$i_D(t) \approx I_D \left(1 + \frac{v_d}{V_T}\right)$$

$$i_D(t) \approx I_D \left(1 + \frac{5 \cdot 2\text{mV}}{25\text{mV}}\right)$$

$$i_D(t) \approx 1.4 \cdot I_D$$

Therefore, current changes by 40%.

(B) Current change limited to 10%

Then,

$$.1 = \frac{v_d \cdot 2}{V_T}$$

$$v_d = 0.05 \cdot 25\text{mV}$$

$$\boxed{v_d = 1.25\text{mV maximum allowable positive or negative voltage}}$$

PROBLEM 4.62

4.62 A 9.1-V zener diode exhibits its nominal voltage at a test current of 20 mA. At this current the incremental resistance is specified as 10 Ω. Find V_{Z0} of the zener model. Find the zener voltage at a current of 10 mA and at 50 mA.

(A) Find V_{Z0}

$-V_Z = 9.1\text{V}$

$-I_{Zr} = 20\text{mA}$

$r_Z = 10\Omega$

$V_{Z0} = V_Z - r_Z I_Z$

$V_{Z0} = -9.1\text{V} + 10\Omega \cdot 20\text{mA} = -9.1 + 0.2$

$\boxed{V_{Z0} = -8.9}$

(B) Find $V_{Z(10\text{mA})}$

$V_Z = -8.9\text{V} - 10\Omega \cdot 10\text{mA}$

$\boxed{V_Z = -9.0\text{V}}$

(C) Find $V_{Z(50\text{mA})}$

$V_Z = -8.9\text{V} - 10\Omega \cdot 50\text{mA}$

$\boxed{V_Z = -9.4\text{V}}$

Problem E4.26

4.26 Assuming the diodes to be ideal, describe the transfer characteristic of the circuit shown in Fig. E4.26.

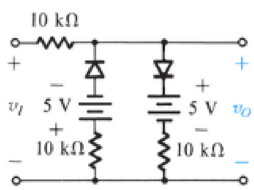
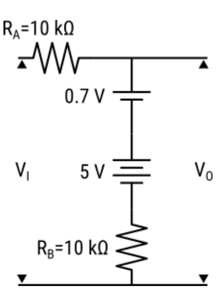


Figure E4.26

Ans. $v_O = v_I$ for $-5 \le v_I \le +5$
$v_O = \frac{1}{2}v_I - 2.5$ for $v_I \le -5$
$v_O = \frac{1}{2}v_I + 2.5$ for $v_I \ge +5$

In this problem, draw equivalent circuit for each input region and explain how the output voltage is derived. Also plot the transfer curve. seminar

(A) $v_I \ge 5.7\text{V}$



$v_I = IR_A + 0.7\text{V} + 5\text{V} + IR_B = I(R_A + R_B) + 5.7$

$$I = \frac{v_{R_B}}{R_B}$$

$$v_{RB} = v_O - 0.7\text{V} - 5\text{V} = v_O - 5.7\text{V}$$

Now we substitute:

$$v_I = \frac{v_{R_B}(R_A + R_B)}{R_B} + 5.7$$

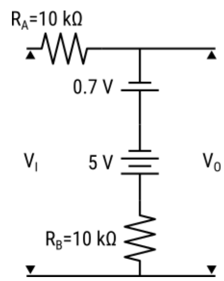
$$v_I = (v_O - 5.7)\frac{R_A + R_B}{R_B} + 5.7$$

Now solve for v_O:

$$v_O = (v_I - 5.7)\frac{R_B}{R_A + R_B} + 5.7$$

$$\boxed{v_O = \frac{1}{2}(v_I - 5.7) + 5.7}$$

(B) $v_I \leq -5.7\text{V}$



$$v_I = IR_A - 0.7\text{V} - 5\text{V} + IR_B = I(R_A + R_B) - 5.7$$

$$I = \frac{v_{R_B}}{R_B}$$

$$v_{RB} = v_O + 0.7\text{V} + 5\text{V} = v_O + 5.7\text{V}$$

Now we substitute:

$$v_I = \frac{v_{R_B}(R_A + R_B)}{R_B} - 5.7$$

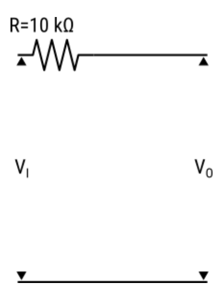
$$v_I = (v_O + 5.7)\frac{R_A + R_B}{R_B} - 5.7$$

Now solve for v_O:

$$v_O = (v_I + 5.7)\frac{R_B}{R_A + R_B} - 5.7$$

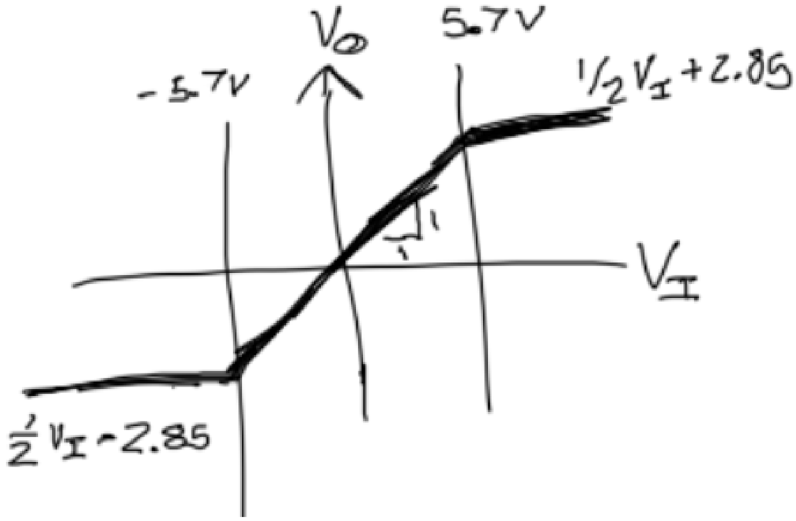
$$\boxed{v_O = \frac{1}{2}(v_I + 5.7) - 5.7}$$

(C) $-5.7 \leq v_I \leq 5.7$



$$\boxed{v_O = v_I}$$

(Transfer Plot)

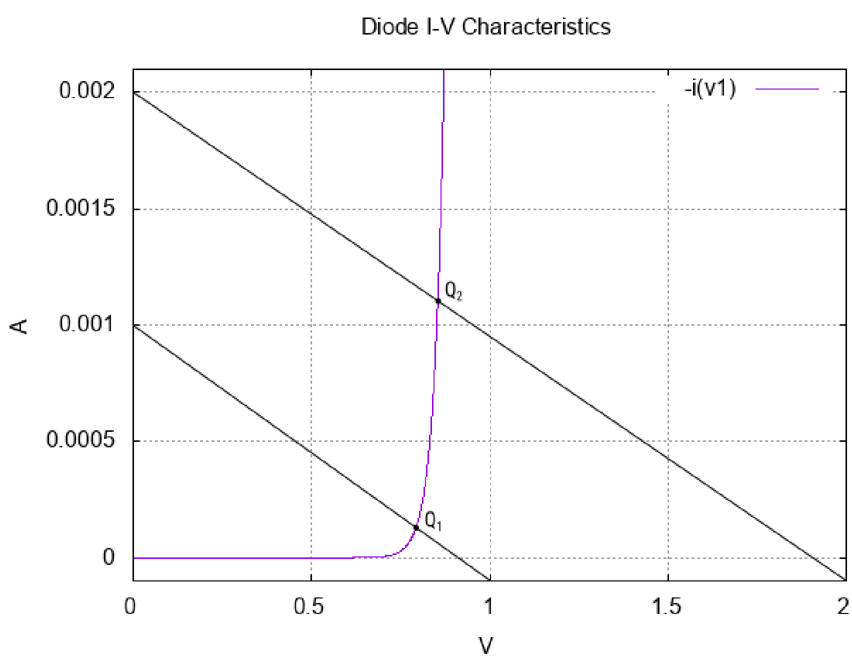


PROBLEM SPICE

(A)

```
.title Diode I-V Characteristics
.model diode d(IS=1.0e-16, N=1.1, temp=20)
v1 1 0
D1 1 0 diode

.control
dc v1 0 2 0.001
set gnuplot_terminal=png/quit
gnuplot $file -i(v1) ylimit 0 2e-3
.endc
.end
```

Plot:



(b) $V_{DD} = 1\text{V}$
Given

```
import numpy
import pint
unit = pint.UnitRegistry()

R = 1 * unit.kohm
v = {'DD': 1 * unit.V, 'D': []}
i = {'S': 10e-16 * unit.A, 'D': []}
```

Assume

$V_T = 25\text{mV}$ (thermal voltage at room temperature).

$V_{D[0]} = 0.7\text{V}$

```
v['T'] = 25 * unit.mV
v['D'].append(0.7 * unit.V)
```

Solve for $I_{D[0]}$ from diode characteristic equation

$$I_{D[0]} = I_S \cdot e^{V_{D[0]}/V_T}$$

```
i['D'].append((i['S']*numpy.exp(v['D'][0] / (v['T']*1.1))).to('mA'))
print("\\noindent\\[I_{D[0]} =", f"{i['D'][0]:.3~Lx}\\]")
```

$$I_{D[0]} = 0.113\,\text{mA}$$

Solve for $I_{D[1]}$ by KVL

$$V_{DD} = I_D R + V_D$$

$$\text{So, } I_D = \frac{V_{DD} - V_D}{R}$$

```
i['D'].append(((v['DD'] - v['D'][0])/R).to('mA'))
print("\\noindent\\[I_{D[1]} =", f"{i['D'][1]:.3~Lx}\\]")
```

$$I_{D[1]} = 0.3\,\text{mA}$$

Iterative solution for V_D and I_D

$$V_{D[n]} - V_{D[n-1]} = V_T \ln \frac{I_{D[n]}}{I_{D[n-1]}}$$

```
iterations = 3

table = [["Iteration", "$V_D$ (V)", "$I_D$ (mA)"],
         [0, f"{v['D'][0].magnitude:.6}", f"{i['D'][0].magnitude:.6}"]]

for n in range(1, iterations+1):
    v['D'].append(v['D'][n-1] + v['T']*numpy.log(i['D'][n]/i['D'][n-1]))
    table.append([n, f"{v['D'][n].magnitude:.6}", f"{i['D'][n].magnitude:.6}"])
    i['D'].append(((v['DD'] - v['D'][n])/R).to('mA'))

print(table)
```

Iteration	V_D (V)	I_D (mA)
0	0.7	0.113441
1	0.724313	0.3
2	0.7222	0.275687
3	0.722391	0.2778

(b) $V_{DD} = 2\text{V}$
Given

```
import numpy
import pint
unit = pint.UnitRegistry()

R = 1 * unit.kohm
v = {'DD': 2 * unit.V, 'D': []}
i = {'S': 10e-16 * unit.A, 'D': []}
```

Assume

$V_T = 25\text{mV}$ (thermal voltage at room temperature).

$V_{D[0]} = 0.7\text{V}$

```
v['T'] = 25 * unit.mV
v['D'].append(0.7 * unit.V)
```

Solve for $I_{D[0]}$ from diode characteristic equation

$$I_{D[0]} = I_S \cdot e^{V_{D[0]}/V_T}$$

```
i['D'].append((i['S']*numpy.exp(v['D'][0] / v['T']/1.1)).to('mA'))
print("\\noindent\\[I_{D[0]} =", f"{i['D'][0]:.3~Lx}\\]")
```

$$I_{D[0]} = 0.113\,\text{mA}$$

Solve for $I_{D[1]}$ by KVL

$$V_{DD} = I_D R + V_D$$

$$\text{So, } I_D = \frac{V_{DD} - V_D}{R}$$

```
i['D'].append(((v['DD'] - v['D'][0])/R).to('mA'))
print("\\noindent\\[I_{D[1]} =", f"{i['D'][1]:.3~Lx}\\]")
```

$$I_{D[1]} = 1.3\,\text{mA}$$

Iterative solution for V_D and I_D

$$V_{D[n]} - V_{D[n-1]} = V_T \ln \frac{I_{D[n]}}{I_{D[n-1]}}$$

```
iterations = 3

table = [["Iteration", "$V_D$ (V)", "$I_D$ (mA)"],
         [0, f"{v['D'][0].magnitude:.6}", f"{i['D'][0].magnitude:.6}"]]
```

```
for n in range(1, iterations+1):
    v['D'].append(v['D'][n-1] + v['T']*numpy.log(i['D'][n]/i['D'][n-1]))
    table.append([n, f"{v['D'][n].magnitude:.6}", f"{i['D'][n].magnitude:.6}"])
    i['D'].append(((v['DD'] - v['D'][n])/R).to('mA'))

print(table)
```

Iteration	V_D (V)	I_D (mA)
0	0.7	0.113441
1	0.760971	1.3
2	0.75977	1.23903
3	0.759794	1.24023

Problem Appendix: Code

```
from __future__ import annotations
import pint
import unittest

unit = pint.UnitRegistry()

def printEquation(tag, value, digits):
    print(f"\\noindent\\[{tag} = {value:.{digits}Lx}\\]")

def printBoxedEquation(tag, value, digits):
    print("\\noindent\\[\\boxed{", f"{tag} = {value:.{digits}Lx}", "}\\]")


class Node:
    def __init__(self, voltage):
        self.voltage = voltage

    @classmethod
    def fromVoltage(cls, voltage):
        return Node(voltage)

    @classmethod
    def toBranchNode(cls, branch, node):
        return Node(node.voltage + branch.drop)

    @classmethod
    def fromNodeBranch(cls, node, branch):
        return Node(node.voltage - branch.drop)

    def __sub__(self, b):
        return self.voltage - b.voltage

class Branch:
    def __init__(self, drop):
        self.drop = drop
        self.current = None

    @classmethod
    def fromVoltage(cls, voltage):
        return Branch(voltage)

    @classmethod
    def fromNodeToNode(cls, nodeA, nodeB):
```

```
        return Branch(nodeA - nodeB)

    def swallowBranch(self, branch):
        newBranch = Branch(self.drop + branch.drop)
        if self.current != None:
            newBranch.current = self.current
        elif branch.current != None:
            newBranch.current = branch.current

        if branch.current == None or self.current == None:
            return newBranch
        else:
            raise RuntimeError("Currents must be the same.")

    def clearCurrent(self):
        self.current = None

    def __add__(self, b):
        return self.current + b.current

    def __radd__(self, other):
        if other == 0:
            return self
        else:
            return self.__add__(other)

    def __neg__(self):
        newBranch = Branch(self.drop)
        newBranch.current = -self.current
        return newBranch

    def setCurrentFromResistance(self, resistance):
        self.current = self.drop/resistance

    def setCurrentTowardsBranches(self, *branches):
        self.current = sum(branches)

class TestBranchAdd(unittest.TestCase):
    def setUp(self):
        self.branchA = Branch.fromVoltage(1)
        self.branchB = Branch.fromVoltage(2)
    def test_currentFromBranchA(self):
        self.branchA.current = 1

        newNode = self.branchA.swallowBranch(self.branchB)

        self.assertEqual(newNode.current, self.branchA.current)

    def test_currentFromBranchB(self):
        self.branchB.current = 1

        newNode = self.branchA.swallowBranch(self.branchB)

        self.assertEqual(newNode.current, self.branchB.current)

    def test_currentIsNone(self):
        newNode = self.branchA.swallowBranch(self.branchB)

        self.assertEqual(newNode.current, None)
```

```python
    def test_currentNotSame_throwsException(self):
        self.branchA.current = 1
        self.branchB.current = 2

        self.assertRaises(RuntimeError, lambda: self.branchA.swallowBranch(self.branchB))

class TestBranchNode(unittest.TestCase):
    def setUp(self):
        self.branch = Branch.fromVoltage(1)
        self.node = Node.fromVoltage(1)

    def test_nodeFromBranchNode(self):
        newNode = Node.toBranchNode(self.branch, self.node)
        self.assertEqual(newNode.voltage, 2)

    def test_nodeToBranchNode(self):
        newNode = Node.fromNodeBranch(self.node, self.branch)
        self.assertEqual(newNode.voltage, 0)

    def test_branchCurrentFromResistance(self):
        self.branch.setCurrentFromResistance(1)
        self.assertEqual(self.branch.current, 1)

if __name__=='__main__':
    unittest.main()
```