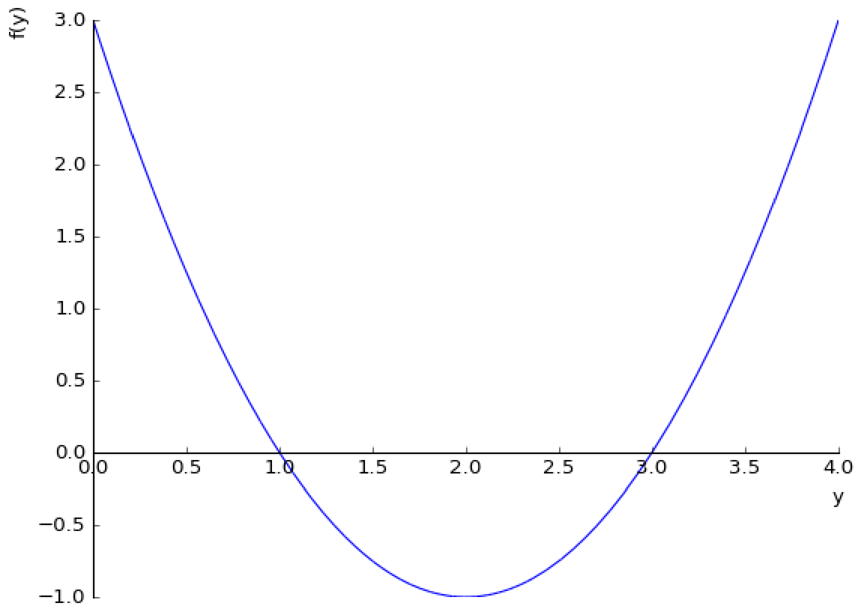
In [1]:
```
from sympy import *
```

Example 1: Identify and classify the equilibrium points of y' = y^2-4y+3. Determine the intervals of increase/decrease and concavity of y.

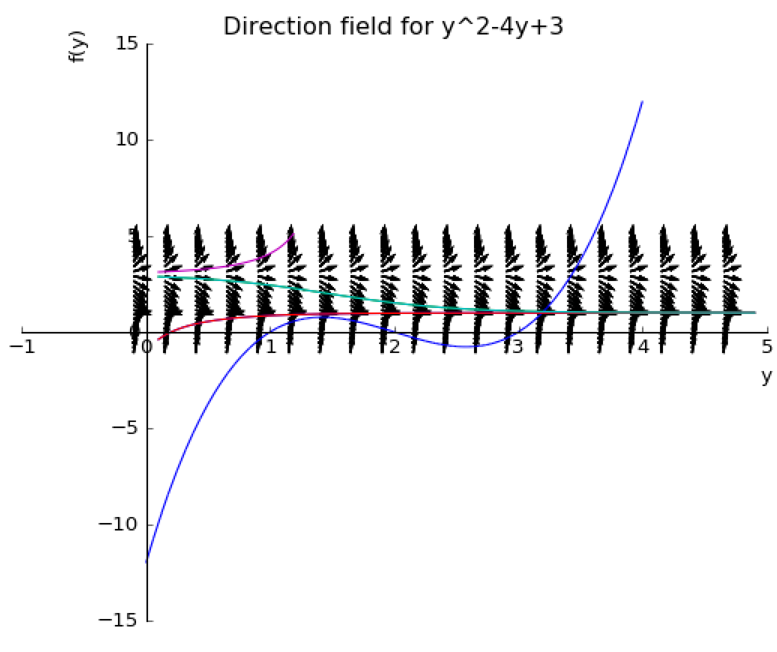
In [2]:
```
matplotlib notebook
```

In [5]:
```
# Find critical values/equilibrium points
y=symbols('y')
f=y**2-4*y+3
eqpts=solve(f,y)
print('The critical values are',eqpts)
print('Based on the information below, 1 is stable and 3 is unstable.')
# Use graph to determine sign of y' = f(y) for y>0
plot(f,(y,0,4))  # chose plot domain to include all equilibrium solutions
print('y is increasing for y in (0,1) and (3,oo) and decreasing for y in (1,3)')
# Differentiate the ODE to get y" = f'(y)*y', or y" = f'(y)*f(y)
ddf=diff(f,y)*f
# Plot y" to determine concavity
plot(ddf,(y,0,4))
print('y"=',ddf)
print('y is concave up for y in (1,2) and (3,oo) and concave down for y in (0,1) and (2,3)')
```

```
The critical values are [1, 3]
Based on the information below, 1 is stable and 3 is unstable.
```




```
y is increasing for y in (0,1) and (3,oo) and decreasing for y in (1,3)
```




```
y"= (2*y - 4)*(y**2 - 4*y + 3)
y is concave up for y in (1,2) and (3,oo) and concave down for y in (0,1) and (2,3)
```

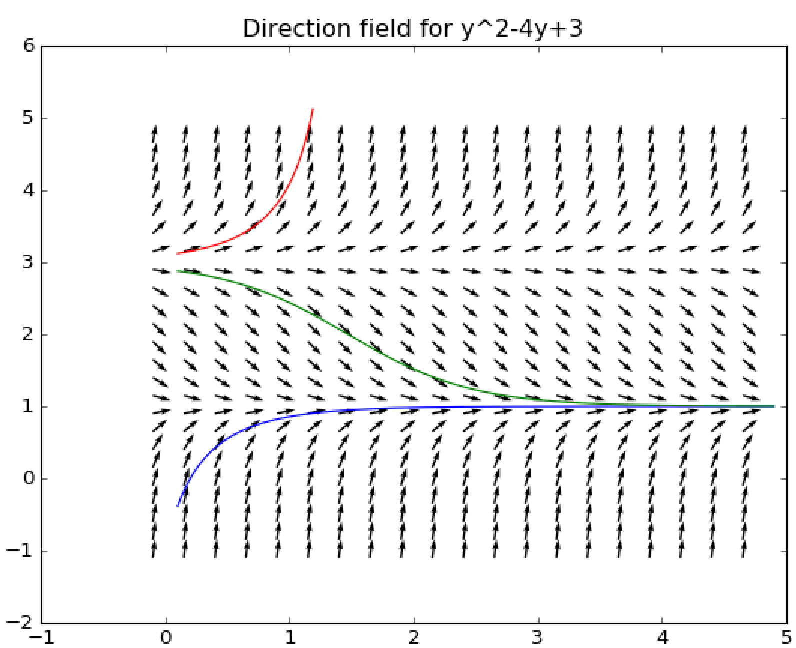
Example 2: Solve the ODE to confirm answers above

In [6]:
```
# solve directly using dsolve
t=symbols('t')
y=Function('y')
deq=diff(y(t),t)-y(t)**2+4*y(t)-3
ysoln=dsolve(deq,y(t))
print('Solution to the ODE is',ysoln)
```

```
Solution to the ODE is y(t) == (3*C1 - exp(2*t))/(C1 - exp(2*t))
```

In [13]:
```
matplotlib notebook
```

In [14]:
```
# Last question on computer assignment shown here
# plot direction field
import numpy as np
import matplotlib.pyplot as plt
T,Y=np.meshgrid(np.arange(-0.1,4.9,0.25),np.arange(-1.1,4.9,0.25))
dYdT=Y**2-4*Y+3
U=1/(1+dYdT**2)**0.5*np.ones(T.shape)
V=1/(1+dYdT**2)**0.5*dYdT
plt.title('Direction field for y^2-4y+3')
Q=plt.quiver(T,Y,U,V)
# Plot Solution Curves
tplot=np.arange(0.1,4.9,0.01)
yplot1=(3*(1/2)-np.exp(2*tplot))/(1/2-np.exp(2*tplot))
plt.plot(tplot,yplot1)
yplot2=(3*(-19)-np.exp(2*tplot))/((-19)-np.exp(2*tplot))
plt.plot(tplot,yplot2)
tplot2=np.arange(0.1,1.2,0.01)
yplot3=(3*21-np.exp(2*tplot2))/(21-np.exp(2*tplot2))
plt.plot(tplot2,yplot3)
```



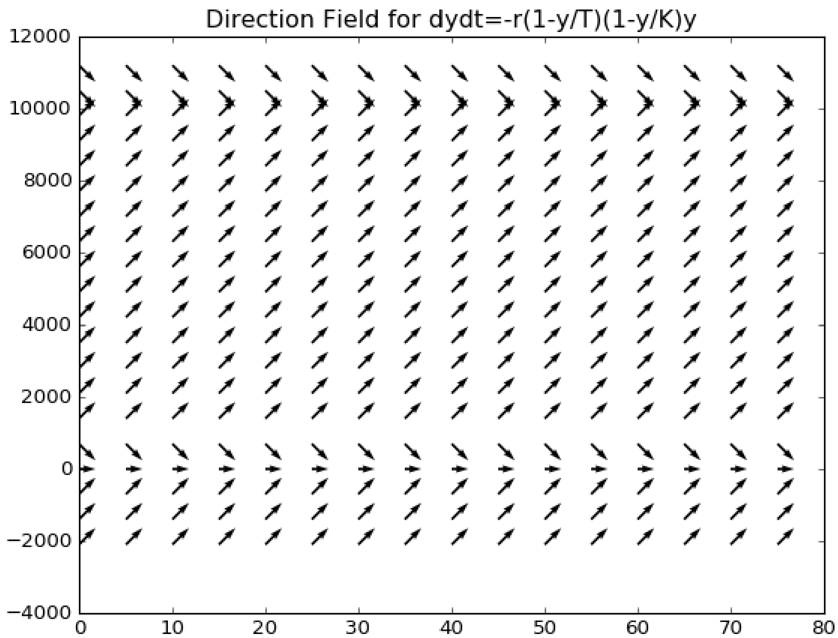
Out[14]:
```
[<matplotlib.lines.Line2D at 0x7fe514985400>]
```

Example 3: y' = -r(1-y/T)(1-y/K)y

In [7]:
```
# Part 1: Direction Field
from numpy import *
import matplotlib.pyplot as plt
```

In [8]:
```
matplotlib notebook
```

In [9]:
```
T, Y = meshgrid(arange(0, 80, 5), arange(-2100, 11900, 700))
dYdT = -0.1*(1-Y/1000)*(1-Y/10000)*Y
U = ones(T.shape)
V = 1/sqrt(1+dYdT**2)*dYdT
plt.figure()
plt.title('Direction Field for dydt=-r(1-y/T)(1-y/K)y')
Q = plt.quiver(T, Y, U, V)
#NOTE the changes in directions around 0, 1000, and 10000
```



In [64]:
```
# Part b/c: Find and classify equilibrium points. Describe
from sympy import *
r,y,T,K=symbols('r y T K')
f=-1*r*(1-y/T)*(1-y/K)*y
eqpts=solve(f,y)
print('The equilibrium points are',eqpts)
print('Based on the direction field above, 0 and K are stable, T is unstable.')
print('If 0<y(0)<T, the population goes to 0. If y(0)>T, the population goes to K.')
```

```
The equilibrium points are [0, K, T]
Based on the direction field above, 0 and K are stable, T is unstable.
If 0<y(0)<T, the population goes to 0. If y(0)>T, the population goes to K.
```

In [ ]: