Name: Larosh Shaikh
Due Date: 12/23/2024

Math 330
Project 1 / Ch 2.2

Dr. Andrew Hofstrand
Due Date: 12/30/2024

Problem 2.2.1: Tangent Lines

a) First, plot the function, $f(x) = ln(x)$, in blue, on the interval $-1 \leq x \leq 6$ and $-2 \leq y \leq 3$.

b) Now find the tangent lines of $f(x)$, using D[], at the points $x = 1$ and $x = e$. Plot the tangent lines in dashed-red and dashed-gray, respectively, on the same domain and range as above.

c) Plot the two points above on $f(x)$ where you evaluated the tangent lines in black.

d) Now overlay each of the plots you made above on a single graph. Make sure you can see everything you plotted.

Problem 2.2.2: Integrating a discontinuous function

a) The so-called Heaviside piece-wise function is defined as follows:

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

Plot this function $G(x) = 2H(x) - 1$

b) Now find the indefinite integral of G(x), using Integrate[] and Piecewise[], and plot the resulting function.

c) Integrate the function you found above another time.

d) Plot all three functions above, including G(x), on a single graph for $-2 \leq x$, $y \leq 2$ in different colors.

e) What do you notice about the continuity and smoothness of the indefinite integrals as you keep integrating G(x)? Conversely, what happens when you go backwards and differentiate each expression?

Solution 2.2.1: Tangent Lines

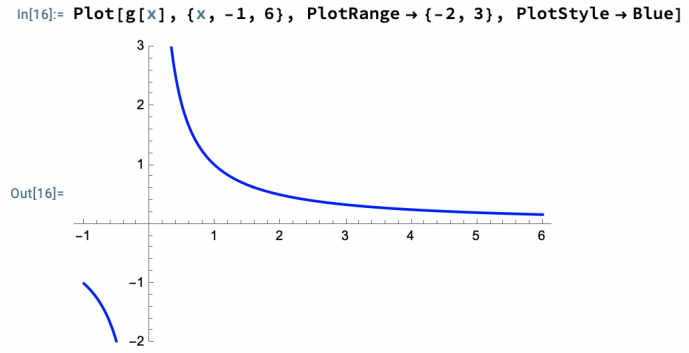
a) In order to plot the function, $f(x) = ln(x)$, we first need to define the function. We can do this by using the following code:

```
f[x_] := Log[x]
```

Now that we have defined the function, we can plot it using the following code:

```
Plot[f[x], {x, -1, 6}, PlotRange -> {{-1, 6}, {-2, 3}}]
```

The resulting plot is shown below:



b) In order to find the tangent lines of $f(x)$, using D[], at the points $x = 1$ and $x = e$, we can do some math ourselves. We know that the derivative of $f(x)$ is $\frac{1}{x}$. We also know that the equation of a line is $y - y_1 = m(x - x_1)$. We can use this information to find the tangent lines of $f(x)$ at the points $x = 1$ and $x = e$.

$$\begin{aligned} y - y_1 &= m(x - x_1) \\ y - 0 &= \frac{1}{1}(x - 1) \\ y &= x - 1 \end{aligned}$$

$$\begin{aligned} y - y_1 &= m(x - x_1) \\ y - 1 &= \frac{1}{e}(x - e) \\ y &= \frac{x}{e} \end{aligned}$$

We can represent these new functions in Mathematica using the following code:

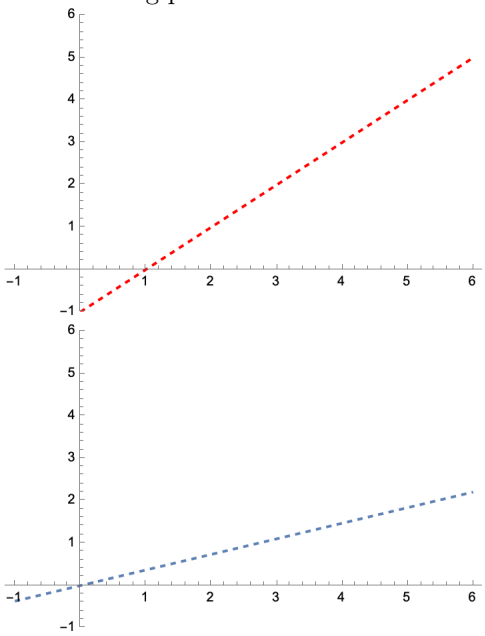
```
g[x_] := x - 1
h[x_] := x/e
```

Now that we have defined the functions, we can plot them individually using the following code:

```
Plot[g[x], {x, -1, 6}, PlotRange -> {-1, 6},
PlotStyle -> {Red, Dashed}]
```

```
Plot[h[x], {x, -1, 6}, PlotRange -> {-1, 6},
PlotStyle -> {Grey, Dashed}]
```

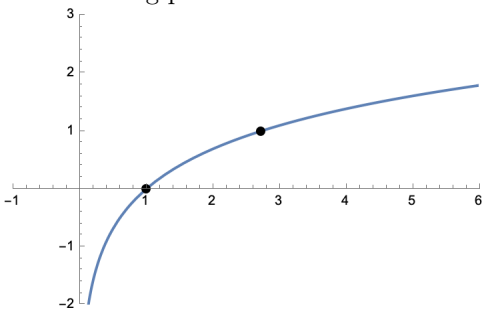
The resulting plots are shown below:



c) Now we just want to plot the two points where I evaulated the lines in black. This is achieveable by:

```
Show[Plot[f[x], {x, -1, 6}, PlotRange -> {{-1, 6}, {-2, 3}}],
Graphics[{PointSize[Large], Point[{1, 0}], Point[{E, Log[E]}]}]]
```

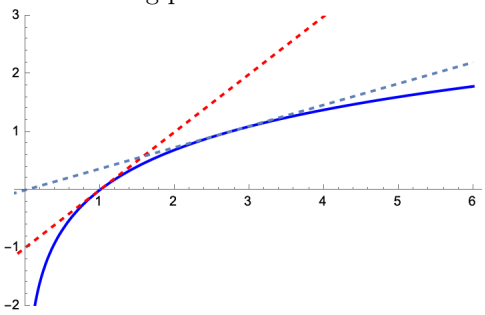
The resulting plot is shown below:



d) Now we just want to overlay each of the plots we made above on a single graph. This is achieveable by:

```
Show[Plot[f[x], {x, -1, 6}, PlotRange -> {{-1, 6}, {-2, 3}}],
Graphics[{PointSize[Large], Point[{1, 0}], Point[{E, Log[E]}]}],
Plot[g[x], {x, -1, 6}, PlotRange -> {-1, 6},
PlotStyle -> {Red, Dashed}],
Plot[h[x], {x, -1, 6}, PlotRange -> {-1, 6},
PlotStyle -> {Grey, Dashed}]]
```

The resulting plot is shown below:

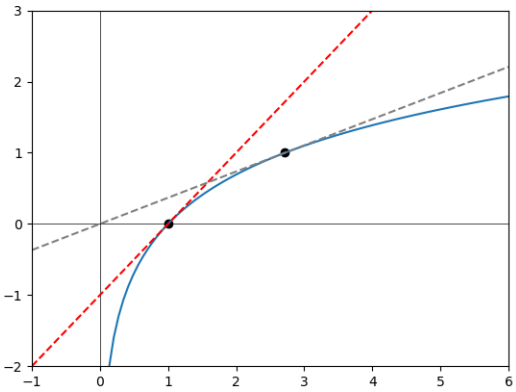


Alternatively, we could use python to plot these points using the matplotlib library. We can do this by using the following code:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-1, 6, 100)
y = np.log(x)
plt.plot(x, y)
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.xlim(-1, 6)
plt.ylim(-2, 3)
plt.scatter(1, 0, color='black')
plt.scatter(np.exp(1), np.log(np.exp(1)), color='black')
y = x - 1
plt.plot(x, y, color='red', linestyle='dashed')
y = x/np.exp(1)
plt.plot(x, y, color='grey', linestyle='dashed')
plt.show()
```

The resulting plot is shown below:



2.2.1 Notes:

- The python code is much more verbose than the Mathematica code. This is because python is a general purpose programming language, while Mathematica is a language specifically designed for mathematical purposes.
- The python code is also much more difficult to read than the Mathematica code.
- A syntaxical note, In Mathematica you may notice the usage of $F[x\_]$ versus $F[x]$, this is to notate the input arguments.
- Another syntaxical note, In Mathematica, you may notice we redfine plot range and the values for which x is plotted twice, x,-1,6 and inside PlotRange. This is because PlotRange is a function that takes in a list of two values, the first being the x range and the second being the y range, but also because PlotRange is defined as where you'd like a plot to extend whereas the x values in the former define where Mathematica evaluates the function.
- There is a rather distinct difference between e and E between versions, please be mindful.

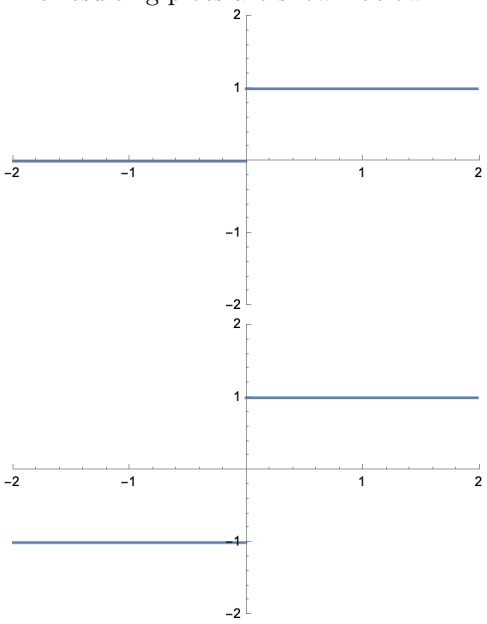
Solution 2.2.2: Integrating a discontinuous function
a) In order to plot the function, $G(x) = 2H(x) - 1$, we first need to define the function. We can do this by using the following code:

```
H[x_] := Piecewise[{{0, x < 0}, {1, x >= 0}}]
G[x_] := 2*H[x] - 1
```

Now that we have defined the function, we can plot it using the following code:

```
Plot[H[x], {x, -2, 2}, PlotRange -> {{-2, 2}, {-2, 2}}]
Plot[G[x], {x, -2, 2}, PlotRange -> {{-2, 2}, {-2, 2}}]
```

The resulting plots are shown below:



b) In order to find the indefinite integral of $G(x)$, we can use Integrate[] and Piecewise[]. We can do this by using the following code:

```
Int1[x_] = Integrate[G[x], x]
```

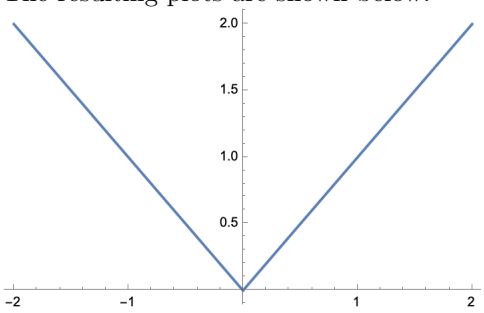
Integrating a piecewise function in Mathematica proves to be very simple. However, integrating a piecewise function by hand is a bit more difficult. We can do this by using the following steps:

$$
\begin{aligned}
G(x) &= 2H(x) - 1 \\
&= 2(0) - 1 \text{ for } x < 0 \\
&= -1 \\
&= 2(1) - 1 \text{ for } x \geq 0 \\
&= 1 \\
&= \int G(x)dx \\
&= \int -1dx \text{ and } \int 1dx \\
G(x) &= -x \text{ for } x < 0 \\
G(x) &= x \text{ for } x \geq 0
\end{aligned}
$$

Now that we have found the indefinite integral of $G(x)$, we can plot it using the following code:

```
Plot[Piecewise[{{-x, x <= 0}}, x], {x, -2, 2}]
```

The resulting plots are shown below:



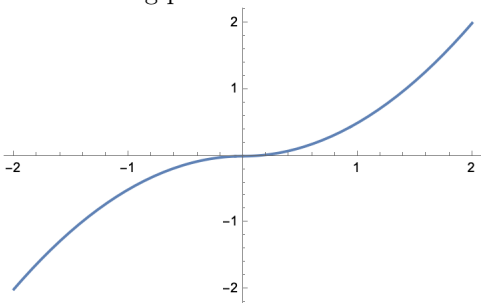
c) In order to integrate the function we found above another time, we can use the following code:

```
Int2[x_] = Integrate[Int1[x], x]
```

Now that we have found the second integral of $G(x)$, we can plot it using the following code:

```
Plot[Piecewise[{{-(x^2/2), x <= 0}}, x^2/2], {x, -2, 2}]
```

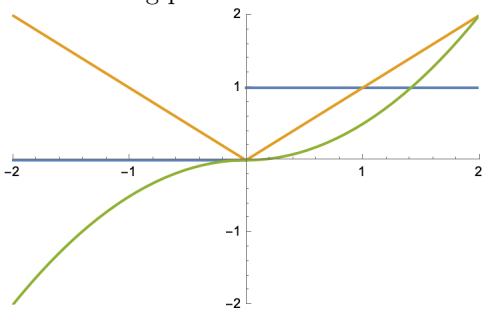
The resulting plots are shown below:



d) In order to plot all three functions above, including $G(x)$, on a single graph for $-2 \leq x, y \leq 2$ in different colors, we can use the following code:

```
Plot[{Piecewise[{{0, x < 0}, {1, x >= 0}}],
Piecewise[{{-x, x <= 0}}, x],
Piecewise[{{-(x^2/2), x <= 0}}, x^2/2]}, {x, -2, 2},
PlotRange -> {{-2, 2}, {-2, 2}}]
```

The resulting plot is shown below:



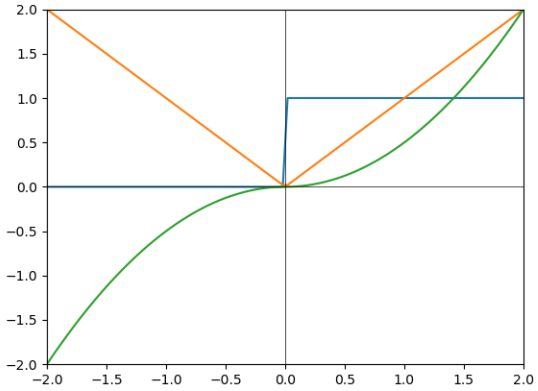
e) As we keep integrating $G(x)$, we notice that the function becomes more and more smooth. Conversely, as we keep differentiating $G(x)$, we notice that the function becomes more and more discontinuous.

Alternatively, we could use python to plot these points using the matplotlib library. We can do this by using the following code:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-2, 2, 100)
y = np.piecewise(x, [x < 0, x >= 0], [0, 1])
plt.plot(x, y)
y = np.piecewise(x, [x < 0, x >= 0], [lambda x: -x, lambda x: x])
plt.plot(x, y)
y = np.piecewise(x, [x < 0, x >= 0], [lambda x: -x**2/2, lambda x: x**2/2])
plt.plot(x, y)
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.xlim(-2, 2)
plt.ylim(-2, 2)
plt.show()
```

The resulting plot is shown below:



2.2.2 Notes:

- We name the integration functions so that we can refer to them later on. I have yet to come up with a more solid naming mechanism, but for now this will have to do.
- You'll notice we do not have to distinguish line colors inside our plot, this is done when plotting 3 seperate functions. Alternatively we could create 3 plots, and use Show[] to combine them, but this is not necessary.