Problem 1

There is my edited output from the program:

```
================================================================================
Dataset: data/chess.csv
--------------------------------
Gain function: train_error()
Average training loss (not-pruned): 0.0
Average test loss (not-pruned): 0.06438127090301003
Average training loss (pruned): 0.021
Average test loss (pruned): 0.06354515050167224
--------------------------------
Gain function: entropy()
Average training loss (not-pruned): 0.0
Average test loss (not-pruned): 0.02508361204013378
Average training loss (pruned): 0.004
Average test loss (pruned): 0.020066889632107024
--------------------------------
Gain function: gini_index()
Average training loss (not-pruned): 0.0
Average test loss (not-pruned): 0.023411371237458192
Average training loss (pruned): 0.004
Average test loss (pruned): 0.019230769230769232
================================================================================
Dataset: data/spam.csv
--------------------------------
Gain function: train_error()
Average training loss (not-pruned): 0.02
Average test loss (not-pruned): 0.20338331410995772
Average training loss (pruned): 0.047
Average test loss (pruned): 0.1987697039600154
--------------------------------
Gain function: entropy()
Average training loss (not-pruned): 0.015
Average test loss (not-pruned): 0.12841214917339486
Average training loss (pruned): 0.028
Average test loss (pruned): 0.12033833141099577
--------------------------------
Gain function: gini_index()
Average training loss (not-pruned): 0.014
Average test loss (not-pruned): 0.1376393694732795
Average training loss (pruned): 0.031
Average test loss (pruned): 0.12226066897347174
```

We could see that comparing to TA's target loss:

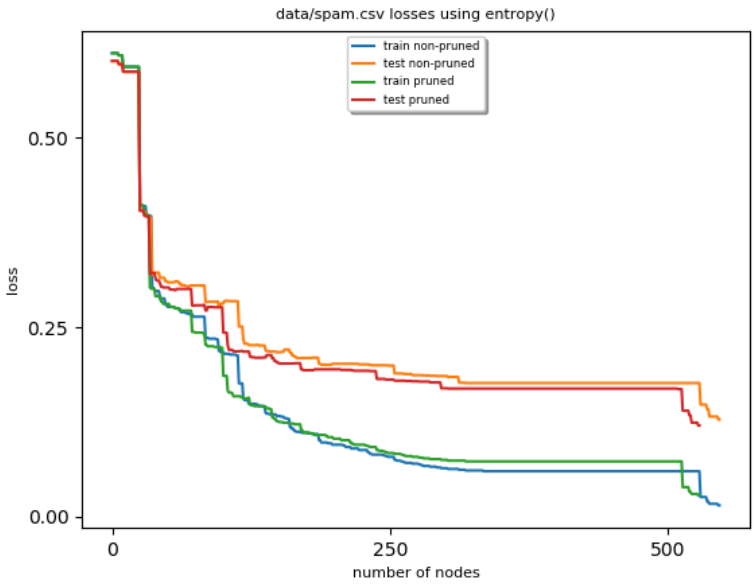
```
Chess dataset
No-pruning Training data loss: 0.0
No-pruning Test data loss: < 0.05
Pruning Training data loss: < 0.03
Pruning Testing loss: < 0.045

Spam Dataset
No-pruning Training data loss: < 0.02
No-pruning Test data loss: < 0.15
Pruning Training data loss: < 0.06
Pruning Testing loss: < 0.11
```

The losses derived from **train_error()** measurement is much higher than that from other two measurements. Which means that using **entropy() and gini_index()** as measurements is better than using **train_error()**.

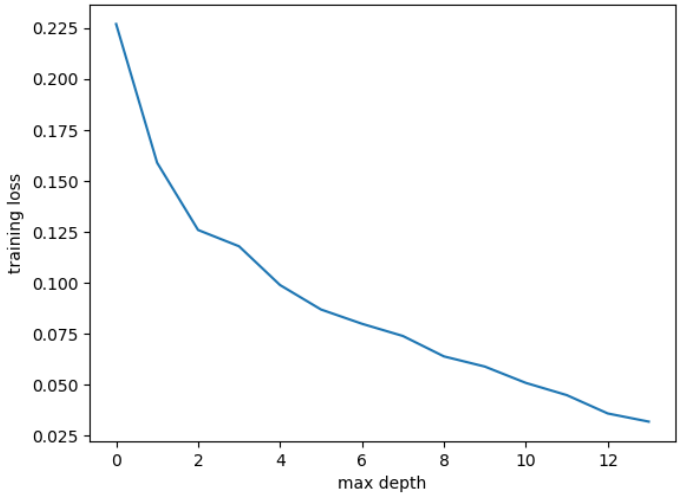
Generally, **traning data loss** increases after applying pruning, which makes sense since the validation dataset altered original tree structure which perfectly fits training data. And **average pruning test losses** are lower than **average no-pruning test losses** which means the pruning implementation is actually working(Figure 1). But these losses still exceeded TA's target limits, meaning that there can be some improvement applied to my pruning part.

Figure 1: Comparison between pruned tree and non-pruned tree



Problem 2

Figure 2: Comparison between different max_depth



From the plots from **Figure 2**, we can see that the curve is convex, meaning it learns fastest at the beginning and then slower. And there's a significantly effective feature that can individually separate the dataset and reach a about 70% accuracy at the beginning.