1. An evil horticulturalist genetically bred three flower species with only one of the following colors each: red, green, and blue. He further designed these flowers so that the colors of the flowers are completely knowable from the width and length of the pedals of the flowers, both ranging from 0 to 5.
2. He designed the flowers in the following way:

If the pedal has width less than 4,

Then if the pedal has length less than 1

Then the color is blue

Else if the pedal has width less than 2

Then the flower is green

Else if the pedal has length greater than 3

Then the flower is green

Else the flower is blue

Else the flower is red.

On a 5 by 5 grid, draw the region colored by the corresponding colors of flowers determined by pedal length and pedal width. Can you write a tree with simpler branches? [NO YOU CAN’T BECAUSE I USED THE GREEDY ALGORITHM]

1. Now consider Jay’s plot. Describe this plot as a decision tree. Try to draw the decision tree with the smallest number of branches and leaves. What do you notice about your way of drawing this tree? [This is one example of the greedy algorithm]
2. Now explain why, in finitely many dimensions, the splits made by a decision tree always separate the region orthogonal to at least one axis, assuming all axes mutually orthogonal.

In other words, explain why cuts introduced by the decision tree can never be ‘slanted’.

2. Two of our brave soldiers broke into the evil horticulturalist’s lab and stole a blueprint of the horticulturalist’s plan. Apparently the horticulturalist once wanted to color the flowers according to these dimensions shown.

1. Create a sampling algorithm for these flowers. Assume that the probability mass is evenly smeared over the 5 by 5 square. Collect your sampling algorithm into a function.
2. Sample 5 sets of 100 points on [0,5] \* [0,5] and train a decision tree on each set of samples, using the decision tree package from data gymnasia. Examine the decision boundaries of your trees. Do the decision boundaries appear close to each other? [They should not because trees are highly unstable.]
3. Now try different values for the number of points sampled. At which point do you consider your tree relatively stable? How might we use a logistic regression to implement this classification? [Decision trees are high-variance classifiers; but in this example they allow for non-global cuts, something not possible for logistic regression; to implement logistic regression, we ought to consider points to the left of x = 4 and to the right.]

3. Unfortunately, one of the soldiers broke a crucial test tube in the lab. Now the horticulturalist would know! In addition, the test tube contained the elixir to make the colors of the flowers cleanly separable by a decision tree. Now regions have flowers with mixed colors!

This has implications for how we grow our tree.

1. Examine the following function in pseudocode and explain what each line is doing. Change a line of the code so that we can examine the predictions given by the tree at any node.

Let $D$ denote data in general. Let $l$ denote the $l$-th feature that we decide to split on, and $m$ the splitting threshold for this feature. Let \*split\* be a function that determines how the decision tree splits. Consider the following function where we pass in a node = Null and depth = 0:

Function grow\_tree(node, data, depth)

Node.prediction = y\_i for any (\vec{x\_i}, y\_i) \in D$

($\bar{l}$, $\bar{m}$, data\_left, data\_right) = split(D).

For all x[$\bar{l}$] <= $\bar{m}$:

Left\_node = grow\_tree(node, data\_left, depth + 1)

For all x[$\bar{l}$] > $\bar{m}$

Right\_node = grow\_tree(node, data\_right, depth + 1)

Return node

[Switch the second line to be mean($y\_i: i\in D$). The second line will cause errors when we have impure regions, for all nodes except the leaves will give conflicting classifications. ]

1. Suppose our split function always produces a previously unseen split that strictly reduces the impurity in each node. Now that the test tube is broken, explain why the decision tree returned by this function will almost always be unsatisfactory. Explain why this may not be a problem when the test tube is not broken––that is, when the colors in the regions are perfectly pure. [It will almost always be overfit for impure data. Overfitting is much less of an issue for perfectly pure data.]
2. Explain how we can use the parameter depth to stop overfitting. Explain why this parameter is not very easy to work with. [There are very few ways to know what the optimal depth to select other than trial and error. When the tree is very deep, using this method alone to overfit is very time consuming.]
3. In addition to the depth parameter, what criterion would you add for the tree to stop splitting? Where would you add a criterion for the tree to stop splitting in the pseudocode above? What parameters should the stopping criterion take? [Add worth\_next\_split function after the \*split\* function. The worth\_next\_split function needs to take in at least the following arguments: data\_left, data\_right, and the cost function. The tree will keep splitting only if the reduction in cost is acceptable. ]

4. To study the products by this accident, we ought to know more about how the tree makes the splits––i.e., we need to look under the function \*split\*’s hood.

(Big idea: information gain is another popular criterion for adding a split to a tree, and, in cases where one of the features has a large number of values, information gain ratio is a better loss function to consider than information gain alone. To be completed in Jay’s notebook.)

1. Our scientists initially considered using information gain as a stopping criterion: if the locally best split does not contribute sufficient information gain, the tree will stop splitting. (Think of something interesting to ask here)
2. But the evil horticulturalist was leaked this information! He responds by vastly adding the dimension of the length of the petals and hence making the flower colors distributed along a thin, long strip. Explain why he adopts this strategy. [The formula for information gain biases it towards features with a lot of distinct values, so we may not gain many useful splits.]
3. Our scientists noticed that the flower’s length dimension grew disproportionately large and noted the evil horticulturalist’s strategy. Our scientists instead started using information gain ratio as a splitting strategy. Explain why our scientists adopted this strategy[Information gain ratio bypasses this problem]