Q1.

(a) Date would be the parameter that would yield least entropy. The value of date is unique for every row in the dataset.

This is not a suitable parameter training as it would amount to overfitting of the input data to the training set only. Each unique date feature would be associated with a unique output and hence the model will only be customized to training data.

(b)

In the above question, the outlook attribute is missing. We can assign a probability to the Outlook attribute and obtain a left and right branch based on the probability of each of the values of outlook to assign the gain. The rest of the tree is built as usual.

Q2.

(b) Accuracy of about 92% is obtained on the IRIS dataset.

(c) A best average accuracy of 91.33% is obtained. The optimum depth of the tree is 2.

Q4.

The performance of the model relative to scikit-learn.

The MSE of my model is using greedy algorithm as 11.34
The MSE of scikit-learn is obtained as 11.04.

Q5.

From the visualization we can infer that the greedy algorithm does not have a fixed length.
It extends as long as there is positive information gain or data remaining in the dataset to be classified.

```
In [51]: show_tree(my_tree)

petal_length>1.9
--->True
    petal_width>1.7
    --->True
        petal_length>4.8
        --->True
            output is virginica
        --->False
            sepal_length>5.9
            --->True
                output is virginica
            --->False
                output is versicolor
    --->False
        petal_length>4.9
        --->True
            petal_width>1.5
            --->True
                sepal_length>6.7
                --->True
                    output is virginica
                --->False
                    output is versicolor
            --->False
                output is virginica
        --->False
            petal_width>1.6
            --->True
                output is virginica
            --->False
                output is versicolor
--->False
    output is setosa
```

Q6.

1. The best possible order of the tree is 4 for best output on training set.
2. The worst possible order for the tree is 1 for the training set.

On the test set we obtain:

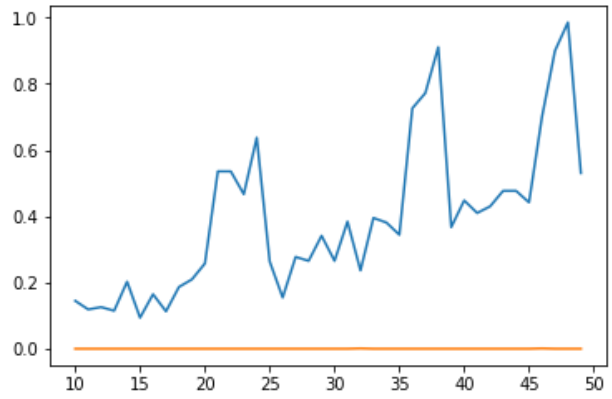
1. Accuracy of 0.9 on greedy and best order trees.
2. Accuracy of 0.66 on the worst possible tree.

Q7.

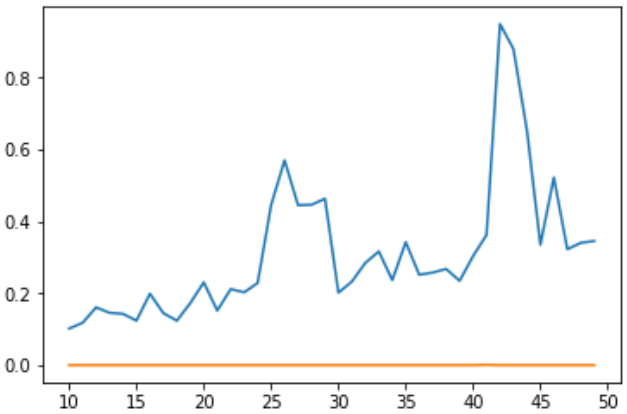
The theoretical time complexity of the algorithm is n-p complete.

The actual graph for training and testing are obtained as follows:

1. The time for tree building increases with M for constant N.
2. The time for execution remains approximately constant with increasing M and N.



For constant n and variable m



For constant m and variable n.

The link to the code is at https://gist.github.com/absdnd/19c03051a3bb2592946ac6a016d735f2

References:

1. https://github.com/random-forests/tutorials/blob/master/decision_tree.ipynb