Questions:

1. **What is a Decision Tree, and how does it make decisions during test time?**

   Answer: A Decision Tree can be expressed as a Boolean function which is a disjunction of conjunctions. It is a type of supervised learning algorithm structured like a tree with decision nodes (tests on features), branches (outcomes from decision nodes) and leaf nodes (predictions).
   During test time, we start at root node with the test sample, traverse through relevant path based on the tests and outcomes captured by the intermittent nodes and conclude when we reach a leaf node with prediction of value/class label.

2. **How does Bagging improve the performance of a Decision Tree?**

   Answer: Bagging involves training multiple Decision Trees on different, randomly picked subsets of the total training dataset, known as bootstrap samples, and finally aggregating the predictions from individual DTs, typically done through majority voting for classification problems and averaging for regression problems.
   As it involves multiple DTs trained from different random subsets of the total training data, the model is less sensitive to the fluctuations in training data, leading to reduced variance and more stable & reliable predictions. It also helps prevent overfitting as individual DTs can be prone to overfitting, especially when they are deep or complex.

3. **In what situations might a Decision Tree overfit the training data, and how can this be mitigated?**

   Answer: A Decision Tree can overfit the training data in situations where there is too small training dataset or too many features relative to number of training samples or if the tree becomes too deep/complex.
   Overfitting can be mitigated through:
   i) Using simple DTs (ex: decision-stump) to the extent possible, that fit the training data reasonably.
   ii) Pruning the tree (pre/post) through early stopping/removing nodes that do not hurt accuracy on validation set much.
   iii) Careful feature selection (avoiding irrelevant ones).
   iv) Setting constraints on minimum number of samples needed per node.

4. **How does Random Forest differ from a single Decision Tree?**

   Answer: In case of a single Decision Tree, we build the tree by using the complete training dataset and go by prediction from this single DT for a new sample. But in Random Forest method, we take only a bootstrapped sub-set of the total training data (Bagging) and also consider only a subset of the features (Random subspace/Feature Bagging) to build a Decision Tree. We repeat this process to build multiple DTs and go with aggregation of the predictions from individual DTs (through maximum voting, averaging etc.) to arrive at the final prediction for a new sample.

5. **What is the main idea behind Boosting in ensemble methods?**

   Answer: In Boosting, the idea is to train weak learners sequentially where each new learner focusses on improving on the errors made by previous ones. This is achieved by training each weak learner on a modified version of data, where incorrectly predicted instances from previous learner are given more weight. Overall accuracy is thus improved through focusing on hard to predict cases.