написать функцию по следующим условиям

1) на вход в функцию посутпает std::vector<double> заполненный уникальными значениями

2) на основании этих данных нужно построить бинарное дерево

3) все элементы вектора из поступивших данных, должны лечь на самый нижний уровень бинарного дерева

т.е. если вам поступило 3 значения то на нижнем уровне будет 3 значения; на уровне над ним будет 2 значения; а на уровне чуть выше будет одно значение, которое и будет корнем всего дерева.

4) все значения для промежуточных элементов должны быть сгенерированны Вами

5) функция должна возвращать указатель на вершину дерева

6) предусмотреть правильное освобождение распределенной под дерево памяти (описать и показать как)

7) проверить результаты своей работы поиском значения по дереву из массивов

std::vector<double> v1{1., 5., 19.};

std::vector<double> v1{1., 17., 19.};

т.е. ваш тестовый код должен пробежаться по дереву из его корня и найти каждое из {1., 5., 19.}

и затем каждое из {1., 17., 19.}

8) проверить работу своей программы на массивах размера более 10

9) написать функцию вывода всех значений из дерева от большего к меньшему.

Всех - значит И тех, что попали к вам в функцию и тех которые вы сами добавляли в дерево,

что бы дерево было сформированно правильно.

10) Неочевидно, однако: Подпись (формат вызова) для функции Вы должны придумать САМИ. И точно так же, сам придумать и реализовать как освобождать распределенную под дерево память.