All codes must be justified (either in a separate file or in comments). Your program must be written in C++ (Python is also allowed). Please ONLY send the .h and .cpp files (no executable)!

1) Implement a data structure supporting the following three operations: **/5**
   a. `void init(const vector<int>& v, int s)`: initialize the data structure content with a fixed n-size vector and an integer s.
      Complexity: O(n*log(n)).
   b. `int set(int i)`: modifies the ith bit (if it exists) in the binary representation of s.
      Complexity: O(1).
   c. `int count()`: returns the number of elements in vector v that are smaller than s.
      Complexity: O(log(n)).

2) In what follows, a u-vector is a vector whose all entries are pairwise different.
   a. Write a function `bool isUVector(const vector<int>& v)` that determines whether a vector is a u-vector.
      Complexity: O(n). **/1**
   b. Write a function `bool isInOrder(const vector<int>& v)` that being given a u-vector v, determines whether it is the inorder of some *balanced* binary search tree.
      Complexity: O(n). **/1**
   c. Write a function `bool isPreOrder(const vector<int>& v)` that being given a u-vector v, determines whether it is the preorder of some binary search tree.
      Complexity: O(n). **/2**
      -- Simpler variant: O(n*log(n)). **/1**
   d. Modify the previous question so that, in the case that v is indeed a preorder, then we also sort this vector.
      Complexity: O(n). **/1**