Algorithms in your answers can be either explained (in plain text), written in pseudo-code or in a programming language amongst C++ or Python. It has no consequence on the notation.

Correctness and Complexity of each algorithm must be justified. If your answer is not justified, then you get at most half of the points. You may also receive up to half of the points if your algorithm is correct but sub-optimal.

The result of a question can be used to answer another question even if you did not manage to answer to it.

Course materials are allowed.

1) Let v be an n-size vector of integers. Show that after a pre-processing in $O(n*\log(n))$ time, we can answer to the following type of queries q(i,j) in $O(\log(n))$-time:
   a. q(i,j): are all elements v[i],v[i+1],…,v[j] pairwise different? **/1**
   b. q(i,j): is the subvector v[i…j] sorted? **/1**
   c. q(i,j): is there some k between i and j such that v[k] is odd? **/1**
   d. q(i,j): are there some distinct k and k' between i and j such that v[k]+v[k']=0? **/1**

2) In what follows, let T be a tree of order n such that we associate to every node x some integer value x.val.
   a. Show that after a pre-processing in $O(n*\log(n))$ time, we can answer to the following type of queries $q_1(x,y,p)$ in $O(\log^2(n))$ time: compute the number of nodes z on the path between x and y so that z.val < p? **/1**
   Hint: use a Heavy-path decomposition…

   b. Compute in $O(n*\log(n))$ time a set S of $O(\sqrt{n})$ nodes such that *every* path of length at least $\sqrt{n}$ must contain at least one node of S. **/1**
   Remark: if T is a path, then we can partition it in $O(\sqrt{n})$ sub-paths, and then we can define S as containing the ends of every sub-path…

   In the next three questions, we consider the following type of queries $q_2(x,y)$: compute the number of pairs (w,z) such that (i) w,z are on the path between x and y, (ii) dist(x,w) < dist(x,z), and (iii) w.val > z.val.

   c. Show that *for a fixed node* x, we can compute in $O(n*\log(n))$ time the values $q_2(x,y)$ *for every node* y. **/1**

   d. Show that after a pre-processing in O(n) time, we can compute $q_2(x,y)$ in $O(\sqrt{n}*\log(n))$ time for every two nodes x and y such that $dist(x,y) \leq \sqrt{n}$. **/1**

   e. Deduce the following result from the four previous questions: after a pre-processing in $O(n*\sqrt{n}*\log(n))$ time, we can compute $q_2(x,y)$ in $O(\sqrt{n}*\log^2(n))$ time for every two nodes x and y. **/1**

3) Recall that, given a sequence $\mathfrak{I}$ of m operations on a disjoint-set data structure, the graph $G(\mathfrak{I})$ has a vertex x for each `makeset(x)` operation, and an edge xy for each `union(x,y)` operation. Show that, if $G(\mathfrak{I})$ is a cycle, then we can execute all m operations in $\mathfrak{I}$ in O(m) time. – Hint: reduce to a sequence $\mathfrak{I}'$ such that $G(\mathfrak{I}')$ is a tree… **/1**