I. Specify whether the following equations hold or not. Give an intuitive justification for your answer. Formal proofs are not needed.

1. $n^3 + 10n^2 = O(\frac{1}{100}n^3 + 100n)$

2. $n^3 \log n = \Omega(n^{2.5} \log^5 n)$

3. $n^3 2^n = O(3^n)$

II. The standard dictionary permits Insert, Delete, and Search operations. We want to permit an additional operation — Delete($\geq x$). This operation deletes all the elements $\geq x$. Outline how this can be implemented as an $O(\log n)$ step algorithm on a balanced tree such as Red-Black tree or a 2,3,4 tree. You don't need to work out all the details. Even if you are not convinced that your implementation runs in $O(\log n)$ steps, just outline your procedure.

III. Consider the following greedy algorithm for finding the minimum spanning tree of an edge-weighted, connected graph G. Index the m edges in decreasing order of weights; i.e. if $i < j$ then $wt(e_i) > wt(e_j)$. Now the algorithm is as follows. (Don't worry whether there is a fast implementation for the algorithm or not.)

For $i = 1 \cdots m$
     if deletion of e_i results in a connected graph, then delete e_i else keep e_i.

Does the resulting tree a minimum spanning tree of the given graph G? Justify your answer by giving a proof if the answer is YES or by giving an example graph on which the algorithm doesn't result in an MST.

IV. Consider the following implementation of the UNION-FIND problem. FIND is implemented by the standard path compression. For any set X, let T_X be its tree. In the usual implementation of UNION(A,B,C), if $size(T_A) \geq size(T_B)$ then the root of T_B is made a child of the root of T_A, else the root of T_A is made a child of the root of T_B. In our implementation, if $size(T_A) \geq \frac{1}{2}size(T_B)$ then the root of T_B is made a child of the root of T_A, else the root of T_A is made a child of the root of T_B.

     Does this implementation of UNION-FIND run in $O(n \log^* n)$ steps? Justify.