Mid-term Exam

Remarks: In all the algorithms, always explain their correctness and analyze their complexity. The complexity should be as small as possible. A correct algorithm with large complexity, may not get full credit.

Choose 5 out of the next 6 questions.

Question 1: In this question, you are given two sorted arrays $A$ and $B$. The median of $A \cup B$ is the element in $A$ or $B$ that is larger than half of the numbers in $A \cup B$ and smaller than the other half. Namely, this is the element that would be in the middle of the array if we concatenate the two arrays and sort the resulting array. Give an algorithm that finds the median of $A \cup B$.

Question 2: Let $T$ be a rooted tree with root $r$. Call a subset of the nodes proper if it does not contain both a node and its parent. Give an algorithm that finds the maximum size proper subset in $T$.

Question 3: This question follows question 2. However, here we assume that each node in the tree has an integer weight $w(v > 0$. The weight $w(S)$ of a subset $S$ of the nodes is $w(S) = \sum_{v \in S} w(v)$. Give an algorithm that finds a proper subset with the maximum possible weight.

Question 4: In this question, you are given an array $A$ and a number $z$. Let $S$ be a sub-array of $A$ (an array containing a part of the elements of $A$). The weight $w(S)$ of $S$ is the sum of the values of the elements in $S$. Give an algorithm that computes the number of sub-arrays of $A$ whose sum is equal to $z$.

Question 5: In this question you are given two sets of positive integers $A = \{a_1, \ldots, a_n\}$, and $B = \{b_1, \ldots, b_m\}$. Assume that $m > n$. A function $f : A \mapsto B$ is said to be a one to one function if $f(a_i) \neq f(a_j)$ for $i \neq j$. Give an algorithm that computes a one to one function $f$ minimizing $\sum_i |f(a_i) - a_i|$.

Question 6: Consider the problem of changing a value $n$ into a sum of minimum possible number of coins. Show that if the coins are powers of some number $c$, namely, $1, c, c^2, c^3, \ldots, c^i$, the greedy algorithm works (recall that the greedy algorithm takes as many coins as possible from $c^i$, then takes as many coins as possible of $c^{i-1}$ and so on).