(1) **Induction**

Let T be the following function: $T(1) = T(2) = T(3) = 1$, and $T(n) = T(n-1) + T(n-2) + T(n-3)$ for all integers $n \geq 4$. Prove that for all positive integers n: $T(n) < 2^n$.

(2) **Logic**

Formalize the following sentences in propositional logic using the provided phrase associated with each prime proposition.

1. Having snow on the ground is necessary for Alice to go skiing.
   - s is "There is snow on the ground."
   - k is "Alice goes skiing."
2. If it is warm out and I don't feel cold, I won't wear my hat.
   - m is "I will wear my hat."
   - w is "It is warm out."
   - c is "I will be cold."

(3) **Modular Arithmetic**

Prove that for all integers n, the number $n^7 - n$ is divisible by 6.

(4) **Dynamic Programming**

Consider the following 3-partition problem. Given integers $a_1, \dots, a_n$, we want to determine whether it is possible to find a partition of $\{1, \dots, n\}$ into three disjoint subsets $I, J, K \subseteq \{1, \dots, n\}$ such that

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_k = \frac{1}{3} \sum_{i=1}^{n} a_i$$

For example, $(2,2,3,4,4,5,7)$ is a YES-instance, because there is the partition $(2,7)$, $(4,5)$, $(2,3,4)$; while $(2,2,3,5)$ is a NO-instance.

1. Let $A = (1/3)\sum_i a_i$. Define a true/false matrix $M[\cdot,\cdot,\cdot]$ of size $A \times A \times (n+1)$ with the meaning that $M[x,y,k]$ is true if and only if there are two disjoint subsets $I, J \subseteq \{1, \dots, k\}$ such that $\sum_{i \in I} a_i = x$ and $\sum_{j \in J} a_j = y$. Which entry represents the answer to the 3-partition problem?
2. Write a recurrence relation to construct M.
3. Give a dynamic programming algorithm for 3-partition that runs in polynomial in n and in $\sum_i a_i$. Prove the correctness and the running time.

(5) **Graphs**

Given an undirected graph $G = (V, E)$ in the adjacency list format, give an algorithm that runs in $O(|V|)$ time and determines whether or not G contains a cycle. The running time of your algorithm should be linear in $|V|$, and not in $|E|$.

(6) **Divide and Conquer**

There are two separate databases. Each database contains n numerical values – so there are $2n$ values total – and assume that no two values are the same. The goal is to determine the *median* of this set of $2n$ values, which is defined as the n-th smallest value. The only way to access these values is through queries to the databases. In a single query, one can specify a value k to one of the two databases, and the chosen database will return the k-th smallest value it contains. Give an algorithm that finds the median using at most $O(\log n)$ queries.