CS30 (Discrete Math in CS), Summer 2021 : Ungraded Practice Problems 2

Due: Not for Submission

Topics: Proofs

---

1 Contradiction

Problem 1 (The Pigeon Hole Principle)**.**

Let n be a positive integer. Suppose there are $n+1$ pigeons residing in n pigeonholes. Then prove there must exist at least one hole with at least two pigeons.

Problem 2.

Prove that $\sqrt{6}$ is irrational.

Problem 3.

Prove that $\sqrt{3}+\sqrt{2}$ is irrational.

Problem 4.

There can be no integers x and y such that $4x^2 = y^2 + 1$.

Problem 5.

Consider the real number $r = a + b\sqrt{2}$ where a and b are rational numbers. Prove that there *cannot* exist a different pair of rational numbers (c, d) such that $r = c + d\sqrt{2}$.

2 Induction

Problem 6.

Prove by induction that $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$.

Problem 7.

Prove by induction that $4^n < n!$ if n is an integer greater than 8.

Problem 8.

Prove by induction that $4^{n+1} + 5^{2n-1}$ is divisible by 21 whenever n is a positive integer.

Problem 9.

Prove that any number $n \geq 12$ can be written as $n = 4x + 5y$ for some *non-negative integers* x and y.

Problem 10.

Prove that any natural number $n \in \mathbb{N}$ can be written as a sum of *one or more, distinct* powers of 2 (note 1 is also a power of 2).

Problem 11.

Consider the following recurrence: $t_1 = 1, t_2 = 3$, and $t_n = t_{\lceil n/2 \rceil} + t_{\lfloor n/2 \rfloor} + 1$ for all $n \geq 3$. Prove that

$$\forall n \in \mathbb{N} : \quad t_n = 2n - 1$$

Problem 12.

Suppose a finite number of players play a round-robin tournament, with everyone playing everyone else exactly once. Each match has a winner and a loser (no ties). We say that the tournament has a *cycle* of length m if there exist m distinct players $(p_1, p_2, \ldots, p_m)$ such that p_1 beats p_2, p_2 beats p_3, $\cdots$, p_{m-1} beats p_m, and p_m beats p_1. Clearly this is possible only for $m \geq 3$.

Prove that if such a tournament has a cycle of length m, for some $m \geq 3$, then it must have a cycle of length *exactly* 3.

Problem 13 (Merge-Sort Correctness)**.** In this exercise, you are going to prove the correctness of MERGE-SORT, an algorithm that you may have seen before to sort an array of numbers.

a. Prove by induction on $n + m$ that the MERGE algorithm given below satisfies the following property: for any $m, n \geq 0$, given two *sorted* (increasing) arrays $X[1 : m]$ and $Y[1 : n]$, MERGE$(X[1 : m], Y[1 : n])$ returns a sorted array containing all elements of X and all elements of Y.

```
procedure MERGE(X[1 : m], Y[1 : n])  ▷ Assumes X, Y are sorted arrays
    ▷ Returns a sorted array containing all elements of X and all elements of Y.
    if n = 0 then:
        return X.
    else if m = 0 then:
        return Y.
        ▷ If the code reaches here then both m and n are > 0.
    else:
        if X[m] > Y[n] then:
            return MERGE(X[1 : m − 1], Y[1 : n]) followed by X[m].
        else: ▷ X[m] ≤ Y[n] here
            return MERGE(X[1 : m], Y[1 : n − 1]) followed by Y[n].
```

b. Prove by induction that MERGESORT takes input an array $A[1 : n]$ and returns a sorted order of the elements of $A[1 : n]$. For this part you may assume MERGE works property (even if you were not able to prove Part (a)).

```
procedure MergeSort(A[1 : n])
    ▷ Returns the sorted order of A[1 : n].
    if n = 1 then:
        return A.
    else:
        m = ⌊n/2⌋.
        L := MergeSort(A[1 : m])
        R := MergeSort(A[m + 1 : n])
        return Merge(L, R).
```

Problem 14. Consider the following implementation of Binary Search in a non-recursive fashion.

```
procedure BinSearch(A[1 : n], x) :   ▷ Assume A is sorted strictly increasing.
    ▷ Returns true if x ∈ A, otherwise returns false.
    L ← 1; U ← n
    while L ≤ U do:
        m ← ⌊(L+U)/2⌋
        if A[m] = x then:
            return true
        else if A[m] < x then:
            L ← m + 1.
        else:
            U ← m − 1
    return false.
```

Prove this program correct by providing

a. The (Pre) and (Post) Conditions.

b. Establish a loop invariant (LI) and prove that it always holds, and on termination implies (Post).

c. Argue that the while loop terminates.

Hint : Take a peek at the solutions to see the (Pre), (Post), and (LI), and then try to prove the rest.

Problem 15.

Suppose you begin with a pile of n stones and split this pile into n piles of one stone each by successively splitting a pile of stones into two smaller piles. For example, if the initial pile has four stones (i.e., $n = 4$), one possibility is:

- split the initial pile with 4 stones into two piles of 2 stones each.
- split one of the piles with 2 stones into two piles with 1 stone each.
- split the other pile with 2 stones into two piles with 1 stone each.

Another possibility is:

- split the initial pile with 4 stones into two piles, one with 3 stones and the other with 1 stone.
- split the pile with 3 stones into one pile with 2 stones and one pile with 1 stone.
- split the pile with 2 stones into two piles with 1 stone each.

Each time you split a pile with $(r+s)$ stones into two piles, one with r stones and one with s stones, you pay rs dollars to the bank. Prove that no matter how you play the game, in the end you *always* pay $\frac{n(n-1)}{2}$ dollars to the bank.

(For example, in the first illustration above, the sum of products is $2 \times 2 + 1 \times 1 + 1 \times 1 = 6$. In the second illustration above, the sum of products is $3 \times 1 + 2 \times 1 + 1 \times 1 = 6$. They are both 6, which is $n(n-1)/2 = 4(4-1)/2$, as stated by the claim I am asking you prove.)

Problem 16 (The Inclusion-Exclusion Formula (Grown up version))**.**

In this problem, $A_1, A_2, \dots, A_n$ are *finite* sets. $[n]$ is a shorthand for the set $\{1, 2, 3 \dots, n\}$. Given any subset $S \subseteq [n]$, $\bigcap_{i \in S} A_i$ is the interesection of the sets named A_i for all $i \in S$. You will be proving the **general inclusion-exclusion formula** which states

$$\text{For any } n \text{ finite sets } A_1, \dots, A_n : \quad \left| \bigcup_{i=1}^{n} A_i \right| = \sum_{S \subseteq [n] : S \neq \emptyset} (-1)^{|S|-1} \left| \bigcap_{i \in S} A_i \right| \qquad \text{(IncExc)}$$

a. Let $A_1, \dots, A_{n+1}$ be a collection of sets. Prove that

$$\left(\bigcup_{i=1}^{n} A_i \right) \cap A_{n+1} = \bigcup_{i=1}^{n} (A_i \cap A_{n+1})$$

b. Prove (IncExc) using mathematical induction.