CS21001 Discrete Structures
Tutorial 2

---

1. Prove the equivalence of the following.

(a) The well-ordering principle of $\mathbb{N}$ (or $\mathbb{N}_0$).
(b) The principle of weak induction.
(c) The principle of the special case of induction.
(d) The principle of strong induction.

2. Let $S \subseteq \mathbb{N}_0 \times \mathbb{N}_0$. It is given that $(0,0) \in S$, and also that whenever $(m,n) \in S$, we have $(m+1,n) \in S$ and $(m,n+1) \in S$. Prove that $S = \mathbb{N}_0 \times \mathbb{N}_0$.

3. What is wrong with the following proof by weak induction?

Theorem: All horses are of the same color.

Proof Let there be n horses. We proceed by induction on n. If $n = 1$, there is nothing to prove. So assume that $n > 1$, and that the theorem holds for any group of $n-1$ horses. From the given n horses discard one, say the first one. Then, all the remaining $n-1$ horses are of the same color by the induction hypothesis. Now, put the first horse back, and discard another, say the last one. Then, the first $n-1$ horses have the same color, again by the induction hypothesis. So all the n horses must have the same color as the ones that were not discarded either time.

4. What is wrong with the following proof by strong induction?

Theorem: $2^n = 1$ for all integers $n \geqslant 0$.

Proof [Basis] For $n = 0$, this is true.

[Induction] Suppose the result holds for $0, 1, 2, \ldots, n-1$. Then, $2^n = 2^1 \times 2^{n-1} = 1 \times 1 = 1$.

5. A finite continued fraction is an expression of the form

$$a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}}$$

with $a_1 \in \mathbb{Z}$, and $a_2, a_3, \ldots, a_n \in \mathbb{N}$. Prove that every rational number a/b with $a \in \mathbb{Z}$ and $b \in \mathbb{N}$ has a finite continued fraction.

6. The game of Nim is played by two players Alice and Bob. There are two piles with m and n sticks. The moves alternate between Alice and Bob. In each move, the player chooses one non-empty pile, and removes one or more sticks from that pile. The player who fails to make the next move loses (that is, the player who makes the last move wins). Alice makes the first move. Prove the following assertions.

(a) If $m = n$, Bob can always win.

(b) If $m \neq n$, Alice can always win.

7. Let a, b be two positive integers, and $d = \gcd(a,b) = ua + vb$ with $u, v \in \mathbb{Z}$. Prove that u and v can be so chosen that $|u| < \dfrac{b}{d}$ and $|v| < \dfrac{a}{d}$.

8. You have coins of denominations 5 and 7. Prove that any integer amount $n \geqslant 24$ can be changed by coins of these denominations.

9. [*Frobenius coin problem*] You have coins of two integral denominations $a, b > 1$ with $\gcd(a,b) = 1$. Prove that any integer amount $n \geqslant (a-1)(b-1)$ can be changed by coins of these two denominations.

10. Let a, b be as in the last exercise. Prove that the amount $(a-1)(b-1) - 1$ cannot be changed by coins of denominations a and b.

11. You have six integers $a_1, a_2, a_3, a_4, a_5, a_6$ arranged in the clockwise fashion on a circle. Their initial values are $1, 0, 1, 0, 0, 0$, respectively. You then run a loop, each iteration of which takes two consecutive integers (that is, (a_1, a_2) or (a_2, a_3) or $\cdots$ or (a_6, a_1)), and increments both the chosen integers by 1. Your goal is to make all the six integers equal. Propose a way to achieve this using the above loop (that is, specify which pairs you choose in different iterations), or prove that this cannot be done.

12. A box contains 100 red marbles, 101 green marbles, and 102 blue marbles. You also have an unlimited external store of marbles of each of these colors. As long as the box contains marbles of at least two different colors, repeat the following: Move two marbles of *different* colors from the box to the store, and then move one marble of the *remaining* color from the store to the box. You stop when all the marbles in the box are of the same color. What is this color?

13. What does the following function return upon the input of two positive integers a, b? Prove it.

```
int f ( int a, int b )
{
   int x, y, u, v;

   x = u = a; y = v = b;
   while (x != y) {
      if (x > y) {
         x = x - y;
         u = u + v;
      } else {
         y = y - x;
         v = u + v;
      }
   }
   return (u + v)/2;
}
```

14. Let A be a sorted array of $n \geqslant 2$ integers with repetitions allowed. Consider the following variant of binary search for x in A. Prove by an invariance property of the loop that the function returns the index of the *first* occurrence of x in A (or -1 if x is not present in A).

```
int first ( int A[], int n, int x )
{
   int L, R, M;

   if ( (A[0] > x) || (A[n-1] < x) ) return -1;
   if (A[0] == x) return 0;
   L = 0; R = n-1;
   while (R - L > 1) {
      M = (L + R + 1) / 2;
      if (A[M] >= x) R = M; else L = M;
   }
   if (A[R] == x) return R;
   else return -1;
}
```

15. Let A and x be as in the last exercise. Write a modified binary-search function that returns the index of the *last* occurrence of x in A (or -1 if the search fails). Prove its correctness.

16. Let A be an array of n integers with $p_1 = A[0] < A[n-1] = p_2$. We want to make an in-place partitioning of A of the form LE_1IE_2G, where the five blocks consist of the following elements of A.

1. L consists of elements less than both p_1 and p_2.
2. E_1 consists of elements equal to p_1.
3. I consists of elements strictly between p_1 and p_2.
4. E_2 consists of elements equal to p_2.
5. G consists of elements greater than both p_1 and p_2.

Propose an algorithm to solve this problem.