The Design and Analysis of Algorithms
National Taiwan University, Fall 2020

Homework 1

Yu-Chieh Kuo B07611039

This homework answers the problem set sequentially.

1. *Find an expression for the sum of the i-th row of the following triangle, which is called the Pascal triangle, and prove the correctness of your claim. The sides of the triangle are 1s, and each other entry is the sum of the two entries immediately above it.*

   Answer:
   Let $T(i)$ be the sum of the i-th row of the Pascal triangle, we have $T(i) = 2^{i-1}$
   <proof>
   In the base case, when $i = 1$, the sum of the first row is 1, which is equal to 2^{1-1}. Assume the sum of row n is 2^{n-1}, then the element of row $n+1$ are each formed by adding two elements of row n, and each element of row n contributes to forming two elements of row $n+1$. Thus, the sum of the $n+1$ row is $2 \cdot 2^{n-1} = 2^n$ as aquired. By induction, we find the expression for the sum of the i-th row of the Pascal triangle.

2. *The Harmonic series $H(k)$ is defined by $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k-1} + \frac{1}{k}$. Prove that $H(2^n) \geq 1 + \frac{n}{2}$, for all $n \geq 0$ (which implies that $H(k)$ diverges).*

   Answer:
   In the basic case, when $n = 0$, $H(2^0) = H(1) = 1 \geq 1$. Assume that $H(2^n) \geq 1 + \frac{2}{n}$ is true, when the case $n+1$,

$$
\begin{aligned}
H(2^{n+1}) &= 1 + \frac{1}{2} + \cdots + \frac{1}{2^n} + \frac{1}{2^n+1} + \cdots + \frac{1}{2^{n+1}} \\
&= H(2^n) + \frac{1}{2^n+1} + \cdots + \frac{1}{2^{n+1}} \\
&\geq (1 + \frac{n}{2}) + \frac{1}{2^n+1} + \cdots + \frac{1}{2^{n+1}} \\
&\geq (1 + \frac{n}{2}) + 2^n \cdot \frac{1}{2^{n+1}} \\
&\geq (1 + \frac{n}{2}) + \frac{1}{2} \\
&\geq 1 + \frac{n+1}{2}
\end{aligned}
$$

   Since the case $n+1$ is still satisfy the inequality, therefore, we prove the Harmonic series inequality by induction.

3. *Consider the following series: 1, 2, 3, 4, 5, 10, 20, 40, ..., which starts as an arithmetic series, but after the first 5 terms becomes a geometric series. Prove that any positive integer can be written as a sum of distinct numbers from this series.*

Answer:
Let $P(n)$ be the proposition to carry out the proof. $P(n)$ is true if positive integer n can be written as a sum of distinct numbers from this series and be false if not. When $1 \leq n \leq 10$, $P(n)$ is true evidently. Assume $n = k > 10$ can be written as a sum of distinct numbers from this series, when $n = k + 1$, let a_i be the largest number in series and be less than $k+1$ simultaneously, that means $a_i < k+1$. Also, notice that $a_i > k + 1 - a_i$ since if $a_i \leq k + 1 - a_i$, it implies $2a_i = a_{i+1} \leq k + 1$ *i.e.* a_{i+1} the largest number in series and be less than $k + 1$ instead of a_i, contradicting to the assumption. Besides, $k > k + 1 - a_i$ implies $P(k + 1 - a_i)$ is true, which means $k + 1 - a_i$ can be written as a sum of distinct numbers from this series $a_{x_1} + a_{x_2} + ... + a_{x_j}$, and $a_i \notin \{a_{x_b} | b = 1, 2, \cdots j\}$. Therefore, $P(k + 1)$ is true.
By induction, we proof that any positive integer can be written as a sum of distinct numbers from this series.

4. *Consider the recurrence relation for Fibonacci numbers $F(n) = F(n-1)+F(n-2)$. Without solving this recurrence, compare $F(n)$ to $G(n)$ defined by the recurrence $G(n) = G(n - 1) + G(n - 2) + 1$. It seems obvious that $G(n) > F(n)$ (because of the extra 1). Yet the following is a seemingly valid proof (by induction) that $G(n) = F(n) - 1$. We assume, by induction, that $G(k) = F(k) - 1$ for all k such that $1 \leq k \leq n$, and we consider $G(n + 1) = G(n) + G(n - 1) + 1 = F(n) - 1 + F(n - 1) - 1 + 1 = F(n + 1) - 1$. What is wrong with this proof?*

   Answer:
   When $n = 1$, $F(1) = 1$ and $G(1) = 1 + 1 = 2$. But in the process of induction, we could see that $G(1) = F(1) - 1$, but obviously it is wrong since $F(1) = 1$ and $G(1) = 2$. Therefore, the proof is wrong with incorrect mathematical induction.

5. *The set of all binary trees that store non-negative integer key values may be defined inductively as follows.*

   (a) *The empty tree, denoted $\perp$, is a binary tree.*

   (b) *If t_l and t_r are binary trees, then $node(k, t_l, t_r)$, where $k \in Z$ and $k \geq 0$, is also a binary tree.*

   So, for instance, $node(2, \perp, \perp)$ is a single-node binary tree storing key value 2 and $node(2, node(1, \perp, \perp), \perp)$ is a binary tree with two nodes — the root and its left child, storing key values 2 and 1 respectively. Pictorially, they may be depicted as below.

   (a) *Define inductively a function SUM that computes the sum of all key values of a binary tree. Let $SUM(\perp) = 0$, though the empty tree does not store any key value.*

   Answer:
   The inductive function SUM is as below:
   $$SUM(node(k, t_l, t_r)) = \begin{cases} k + SUM(t_l) + SUM(t_r) & \text{tree is non-empty} \\ 0 & \text{tree is } \perp \end{cases}$$

   (b) *Suppose, to differentiate the empty tree from a non-empty tree whose key values sum up to 0, we require that $SUM(\perp) = -1$. Give another definition for SUM that meets this requirement; again, induction should be used somewhere in the definition.*

Answer:
The inductive function SUM is as below:

$$SUM'(node(k, t_l, t_r)) = \begin{cases} k + SUM'(t_l) + SUM'(t_r) & \text{tree is non-empty} \\ 0 & \text{tree is } \perp \end{cases}$$

(c) *Define inductively a function $MBSUM$ that determines the largest among the sums of the key values along a full branch from the root to a leaf. Let $MBSUM(\perp) = 0$, though the empty tree does not store any key value*

Answer:
The inductive function $MBSUM$ is as below:

$$MBSUM(node(k, t_l, t_r)) = \begin{cases} k + max(MBSUM(t_l), MBSUM(t_r)) & \text{tree is non-empty} \\ 0 & \text{tree is } \perp \end{cases}$$

Note that the function max will return the maximum value in parameters.

(d) *Suppose, to differentiate the empty tree from a non-empty tree whose key values on every branch sum up to 0, we require that $MBSUM(\perp) = 1$. Give another definition for $MBSUM$ that meets this requirement; again, induction should be used somewhere in the definition.*

Answer:
The inductive function $MBSUM$ is as below:

$$MBSUM(node(k, t_l, t_r)) = \begin{cases} k + max(MBSUM'(t_l), MBSUM'(t_r)) & \text{tree is non-empty} \\ -1 & \text{tree is } \perp \end{cases}$$

Note that the function max will return the maximum value in parameters.