Homework 1

蘇俊杰、劉韋成、曾守瑜

Question1

(2.10) 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.

$$\begin{array}{ccccccccc} & & & & 1 & & & & \\ & & & 1 & & 1 & & & \\ & & 1 & & 2 & & 1 & & \\ & 1 & & 3 & & 3 & & 1 & \\ 1 & & 4 & & 6 & & 4 & & 1 \end{array}$$

Question1

[Claim] The sum of the i-th row of the Pascal triangle is 2^{i-1}
[Base Case] (i=1) $2^{1-1} = 1$
[Inductive Step]
Let the elements of k^{th} row be $a_1, a_2, ..., a_k$ and the sum of k^{th} row be 2^{k-1}.

The sum of $(k+1)^{th}$ row
$= a_1 + (a_1 + a_2) + (a_2 + a_3) + ... + (a_{k-1} + a_k) + a_k$
$= 2 * (a_1 + a_2 + ... + a_{k-1} + a_k)$
$= 2 * 2^{k-1}$
$= 2^k$
$= 2^{(k+1)-1}$

Question2

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).

Question2

[Base Case] (n=0) $H(2^0) = H(1) = 1 \ \geq \ 1 + \frac{0}{2} = 1$

[Induction Step] $H(2^{n+1}) = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{2^n} + \frac{1}{2^n+1} + ... + \frac{1}{2^{n+1}}$

$$\geq 1 + \frac{n}{2} + (\frac{1}{2^n+1} + \frac{1}{2^n+2} + ... + \frac{1}{2^{n+1}})$$

$$\geq 1 + \frac{n}{2} + (\frac{1}{2^{n+1}} + \frac{1}{2^{n+1}} + ... + \frac{1}{2^{n+1}})$$

$$= 1 + \frac{n}{2} + (2^n \ * \frac{1}{2^{n+1}})$$

$$= 1 + \frac{n}{2} + \frac{1}{2}$$

$$= 1 + \frac{n+1}{2}$$

Question3

(2.14) 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.

Question3

Proposition : Any positive integer can be written as a sum of distinct numbers from this series.

Any positive integer can be present as :
10n(十位數) + r(個位數)

and it is easy to prove that any r can be written as a sum of distinct numbers from the series.
ex: when n = 0, 1,2,3,4,5,(5+1),(5+2),(5+3),(5+4)
About n = 1 :
ex: when n = 1, 10+1, 10+2, ..., 10+(5+4)

Question3(Continue)

The thing we have to prove is that :
any 10n can be written as a sum of distinct numbers from infinity series S $= \{10, 20, 40...\}$

The proof is by induction on n.(十位數)

Question3(Continue)

[Base case] n = 0, 1 :
can be proved intuitively (1-19)

[Strongly Inductive Hypothesis]
for all x, $1 \leq x \leq n, x \in N$
can be written as a sum of distinct numbers S_x,
which is a subset of S.

Question3(Continue)

[Induction Step] 10(n + 1) in two cases.
[n + 1 is odd]
let x $= \frac{n}{2}$, $1 \leq x \leq n, x \in N$

10x can be written as a sum of distinct numbers :
$S_x = \{x_1, x_2, ..., x_i\}$ with each element in S.(inductive hypothesis)

10n can be written as a sum of distinct numbers :
$S_n = \{2x_1, 2x_2, ..., 2x_i\}$. Each element is still in S and S_n does not contain $\{10\}$

10(n+1) can be written as a sum of distinct numbers :
$S_{n+1} = \{10, 2x_1, 2x_2, ..., 2x_i\}$

Question3(Continue)

[n + 1 is even]
let x = $\frac{n+1}{2}$, $1 \leq x \leq n, x \in N$

10x can be written as a sum of distinct numbers :
$S_x = \{x_1, x_2, ..., x_i\}$ with each element in S.(inductive hypothesis)

10(n+1) = 2x can be written as a sum of distinct numbers :
$S_{n+1} = \{2x_1, 2x_2, ..., 2x_i\}$ and each element is still in S.

10(n+1) can be written as a sum of distinct numbers from S.

By M.I., we proved the proposition.

Question4

(2.37) 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 $G(n+1)$:

$$G(n+1) = G(n) + G(n-1) + 1 = F(n) - 1 + F(n-1) - 1 + 1 = F(n+1) - 1$$

What is wrong with this proof?

Question4

[Base Case] n=1

We can find that :

$F(1) = 1$ and $G(1) = 1 + 1 = 2$
$G(1) \neq F(1) - 1$

Question4(Continue)

But this question does not define G(1) and G(2)!

So if we define
$G(1) = 0$, $G(2) = 0$ so that $G(3) = 1$, $G(4) = 2$ and so on,
we will find the assumption in this question

$G(1) \neq F(1) - 1$

is exactly true!

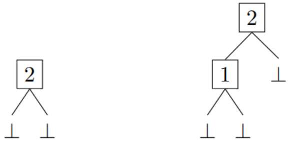
Question5

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 repsectively. Pictorially, they may be depicted as below.



Question5

(a) (5 points) 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.

(b) (5 points) 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.

(c) (5 points) 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.

(d) (5 points) 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.

Question5

- (a) **Inductively** define a function *SUM*
  $\rightarrow$ Define a recursive function
  Base case of recursion: $SUM(\perp) = 0$
  If tree is in the form $node(l, t_l, t_r)$,
  $SUM(node(k, t_l, t_r)) = k + SUM(t_l) + SUM(t_r)$

$$SUM(tree) = \begin{cases} 0, & tree = \perp \\ k + SUM(t_l) + SUM(t_r), & tree = node(k, t_l, t_r) \end{cases}$$

Can I Write Pseudocode?

教授說不行，除非題目特別指定要寫 pseudocode
作業一第五題寫 code 或 pseudocode 我不會扣分
作業二第一題就會扣了，因為教授上課應該有提醒過

Question5

- (b) Base case: $SUM(\perp) = -1$
  除此之外，其他樹的運算結果需要和 (a) 一樣
  問題是，每棵樹都有很多空的子樹，不能讓這些 -1 影響結果
  換句話說，對其他非空的樹而言，它們的 Base case 不應該是上面這條，而是 $SUM(node(k, \perp, \perp)) = k$

$$SUM(tree) = \begin{cases} -1, tree = \perp \\ k, tree = node(k, \perp, \perp) \\ k + SUM(t_l), tree = node(k, t_l, \perp) \ and \ t_l \neq \perp \\ k + SUM(t_r), tree = node(k, \perp, t_r) \ and \ t_r \neq \perp \\ k + SUM(t_l) + SUM(t_r), otherwise \end{cases}$$

Question5

- (b) 這樣寫，更簡潔！

$$SUM(tree) = \begin{cases} -1, & tree = \perp \\ SUM'(tree), & otherwise \end{cases}$$

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

Question5

- c. Define a function *MBSUM*
  從樹根到葉子形成的路徑，總和最大者
  Base case: $MBSUM(\perp) = 0$
  If tree is in the form $node(k, t_l, t_r)$,
  $MBSUM(node(k, t_l, t_r)) = k + \max(MBSUM(t_l), MBSUM(t_r))$

Question5

- (d) Base case: $MBSUM(\bot) = -1$
  除此之外，其他樹的運算結果需要和 (c) 一樣

$$MBSUM(tree) = \begin{cases} -1, tree = \bot \\ k, tree = node(k, \bot, \bot) \\ k + MBSUM(t_l), tree = node(k, t_l, \bot) \text{ and } t_l \neq \bot \\ k + MBSUM(t_r), tree = node(k, \bot, t_r) \text{ and } t_r \neq \bot \\ k + \max(MBSUM(t_l) + MBSUM(t_r)), otherwise \end{cases}$$