NOTE TO STUDENTS: This file contains sample solutions to the term test together with the marking scheme and comments for each question. Please read the solutions and the marking schemes and comments carefully. Make sure that you understand why the solutions given here are correct, that you understand the mistakes that you made (if any), and that you understand *why* your mistakes were mistakes.

Remember that although you may not agree completely with the marking scheme given here it was followed the same way for all students. We will remark your test only if you clearly demonstrate that the marking scheme was not followed correctly.

Question 1. [8 MARKS]

Prove, by complete induction, that $\forall n \in \mathbb{N}, n \geq 1$, n can be expressed as a sum of distinct powers of 2. That is, $n = 2^{i_1} + 2^{i_2} + ... + 2^{i_k}$, for some $k \in \mathbb{N}$, $i_j \in \mathbb{N}, 0 \leq j \leq k$, and all i_j are distinct.

Your proof must clearly indicate any necessary base case(s), the inductive hypothesis, and where the hypothesis is used. Proofs not using complete induction will not receive full marks.

Hint: In the case where n is not a power of 2, consider removing the largest power of 2 less than n.

SAMPLE SOLUTION:

Inductive Step: Let $n \in \mathbb{N}, n \geq 1$.

- **Assume $\boldsymbol{H(n)}$:** $\forall i \in \mathbb{N},\ 1 \leq i < n$, i can be expressed as a sum of distinct powers of 2.
- **Show $\boldsymbol{H(n) \rightarrow C(n)}$:** n can be expressed as a sum of distinct powers of 2.
- **Case:** $n = 2^k$ for some $k \in \mathbb{N}$ (ie. n is a power of 2). Then, the claim holds, as n can be represented as the sum of itself.
- **Case:** $n \neq 2^k, \forall k \in \mathbb{N}$. Choose the largest $l \in \mathbb{N}$ s.t. $2^l < n$. Let $p = n - 2^l$. By $H(n)$, and because $n \geq 3$ so $1 \leq p < n$, p can be written as a sum of distinct powers of 2. By the choice of l, all of those powers of 2 are smaller than 2^l (otherwise, we would have chosen a larger l). Then n can be written as the sum of the powers of 2 that sum to p, plus 2^l, and $C(n)$ holds.

Base Case: No explicit base case is required.

Conclusion: By complete induction, $\forall n \in \mathbb{N}, n \geq 1$, n can be expressed as a sum of distinct powers of 2.

MARKING SCHEME:

- **Structure** [4 marks]:

  (1 mark) Explicit base case or all cases covered in inductive step

  (1 mark) Contains an inductive hypothesis.

  (1 mark) Contains an inductive step.

  (1 mark) Inductive step makes use of inductive hypothesis.

  It is OK if none of these are labelled explicitly.

- **Content** [4 marks]:

  (1 mark) Base case is covered (OK if duplicates 1 and 2)

  (1 mark) Correct inductive hypothesis

  (2 marks) Correct proof in inductive step

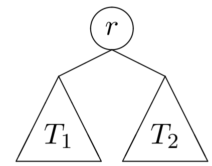
MARKER'S COMMENTS:

- **BC**: Missing base case or base case not covered in proof
- **IH**: Missing inductive hypothesis
- **IS**: Missing inductive step
- **USE, UH**: Does not use inductive hypothesis in inductive step
- **CH**: Incorrect inductive hypothesis
- **CI**: Incorrect proof of inductive step
- **COMP**: Does not use complete induction (-2)
- **D1**: Sum is not distinct (-1 for one occurrence, -2 for twice)
- **IS1**: Does not say why $n - 2^l$ can be made by distinct powers of 2

Question 2. [6 MARKS]

Consider the following recursive definition of *binary trees*.

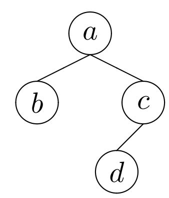
- The *empty tree* (that contains **no** node and **no** edge) is a binary tree.
- If T_1 and T_2 are binary trees and r is a single node, then the tree obtained by adding an edge from r to the root of T_1 (making T_1 the left subtree of r) and by adding an edge from r to the root of T_2 (making T_2 the right subtree of r) is also a binary tree. (This is illustrated on the right.)
- Nothing else is a binary tree.



Consider the *height* h of a binary tree T, defined as follows:

$$h(T) = \begin{cases} 0 & \text{if } T \text{ is empty} \\ 1 + max(h(T_L), h(T_R)) & \text{otherwise, } T_L \text{ and } T_R \text{ are the left and right subtrees of } T \end{cases}$$

For example, the height of the binary tree pictured on the right is 3. The height of the binary tree rooted at d is $1 = 1 + max(0, 0)$, rooted at c is $2 = 1 + max(1, 0)$, rooted at b is $1 = 1 + max(0, 0)$, and rooted at a is $3 = 1 + max(1, 2)$.



Use structural induction to prove that for all binary trees T, the **number of edges** in $T \leq 2^{h(T)} - 2$. Your proof must clearly indicate any necessary base case(s), the inductive hypothesis, and where the hypothesis is used. Proofs not using structural induction will not receive full marks.

SAMPLE SOLUTION:

Note: Due to the error in the question, we did give full marks to proofs that are not fully correct, but had the right components and good reasoning.

Let $e(T)$: the number of edges in T.

Base Case: T is a single node with empty left and right children. Then $e(T) = 0 \leq 0 = 2^1 - 2 = 2^{h(T)} - 2$. So, the base case holds.

Inductive Step: Let T_1 and T_2 be non-empty binary trees. Let r be a node. Let T be a binary tree rooted at r either with T_1 as its left child and T_2 as its right child, or with T_1 as one child and an empty binary tree as the other.

Assume $\boldsymbol{H(\{T_1, T_2\})}$: $e(T_1) \leq 2^{h(T_1)} - 2$ and $e(T_2) \leq 2^{h(T_2)} - 2$.

Show $\boldsymbol{H(\{T_1, T_2\}) \rightarrow C(\{T_1, T_2\})}$: $e(T) \leq 2^{h(T)} - 2$

Case: One child of T is an empty binary tree. By definition, the other child of T is T_1.

$$\begin{aligned} e(T) &= e(T_1) + 1 && \text{add one edge to non-empty child} \\ &\leqslant 2^{h(T_1)} - 2 + 1 && \text{by } H(\{T_1, T_2\}) \\ &\leqslant 2 * 2^{h(T_1)} - 2 && \text{heights are natural numbers, so } 2^{h(T_1)} + 1 \leq 2 * 2^{h(T_1)} \\ &= 2^{1+h(T_1)} - 2 && \\ &= 2^{h(T)} - 2 && \text{by definition of height} \end{aligned}$$

Case: Both children of T are non-empty.

$$\begin{aligned} e(T) &= e(T_1) + e(T_2) + 2 && \text{add one edge to each non-empty child} \\ &\leqslant 2^{h(T_1)} - 2 + 2^{h(T_2)} - 2 + 2 && \text{by } H(\{T_1, T_2\}) \\ &\leqslant 2 * 2^{max(h(T_1), h(T_2))} - 2 \\ &= 2^{1+max(h(T_1), h(T_2))} - 2 \\ &= 2^{h(T)} - 2 && \text{by definition of height} \end{aligned}$$

Conclusion: By structural induction, for all **non-empty** binary trees T, the number of edges in $T \leq 2^{h(T)} - 2$.

MARKING SCHEME:

- **Structure** [4 marks]:
  - (1 mark) Contains a base case
  - (1 mark) Contains an inductive hypothesis in terms of binary tree elements (not n)
  - (1 mark) Contains an inductive step that makes use of the inductive hypothesis
  - (1 mark) Inductive step follows recursive definition of binary trees
- **Content** [2 marks]:
  - (1 mark) Inductive hypothesis is used correctly
  - (1 mark) Proof follows correct reasoning (even if not correct)

MARKER'S COMMENTS:

- **BC**: Missing base case or base case not covered in proof
- **BT**: Missing inductive hypothesis in terms of binary tree elements (not n)
- **IH**: Missing inductive step that uses inductive hypothesis
- **RE**: Induction does not follow recursive definition of binary trees
- **IH2**: Incorrect use of inductive hypothesis
- **CR**: Incorrect reasoning in proof

Question 3. [4 MARKS]

Find a closed form for the recurrence $T(n)$ using repeated substitution (unwinding). Show your work.

$$T(n) = \begin{cases} 1 & \text{if } n = 0 \\ 4 & \text{if } n = 1 \\ 2T(n-1) - T(n-2) & \text{if } n > 1 \end{cases}$$

SAMPLE SOLUTION:

$$\begin{aligned} T(n) &= 2T(n-1) - T(n-2) \\ &= 2(2T(n-2) - T(n-3)) - T(n-2) \\ &= 3T(n-2) - 2T(n-3) \\ &= 3(2T(n-3) - T(n-4)) - 2T(n-3) \\ &= 4T(n-3) - 3T(n-4) \end{aligned}$$

After i substitutions, $T(n) = (i+1)T(n-i) - iT(n-i-1)$.

The base case is reached at $T(1)$, that is, after $n-1$ substitutions.

So $T(n) = (n-1+1)T(1) - (n-1)T(0) = 4n - (n-1) = 3n+1$.

MARKING SCHEME:

- (1 mark) Starts with $T(n) = 2T(n-1) - T(n-2)$ in unwinding
- (1 mark) Applies repeated substitution
- (1 mark) Identifies correct pattern after some number of substitutions
- (1 mark) Correct closed form

- 3 marks for getting correct answer but not using unwinding
- 2 marks for correctly applying characteristic equation but not getting correct answer

MARKER'S COMMENTS:

- **NP**: No need to prove (no deduction of marks)
- **IP**: Incorrect pattern identified (-1 mark, but likely also -1 for WCF)
- **WCF**: Wrong closed form (-1 mark)
- **UU**: Does not use unwinding (-1 mark)