Worth: 8% **Due:** *Before* 10pm on Tuesday 31 January 2012.

Remember to write the *full name*, *student number*, and *CDF/UTOR email address* of *each group member* prominently on your submission.
Please read and understand the policy on Collaboration given on the Course Information Sheet. Then, to protect yourself, list on the front of your submission **every** *source of information you used to complete this homework (other than your own lecture and tutorial notes, and materials available directly on the course webpage). For example, indicate clearly the* **name** *of every student with whom you had discussions (other than group members), the* **title** *of every additional textbook you consulted, the* **source** *of every additional web document you used, etc.*

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions.

---

1. In graphics programming, the following standard trigonometric identities are often used to rotate objects.

$$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$$
$$\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y)$$

Use these formulas to prove that $\forall n \in \mathbb{Z}^+, \forall x \in \mathbb{R}$,

$$(\cos(x) + i\sin(x))^n = \cos(nx) + i\sin(nx).$$

Recall that i is the "imaginary unit", *i.e.*, the square root of -1 (so $i^2 = -1$).

Do *not* convert this problem to exponential notation and try to solve it in that form!

HINT: This is actually simpler than it looks. If your equations seem very complicated, it is most likely because of a mistake in your algebra — everything is supposed to simplify nicely.

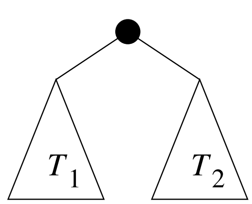
2. Define a function f recursively, as follows:

$$\begin{aligned} f(1) &= 1, \\ f(2) &= 5, \\ f(n) &= 5f(n-1) - 6f(n-2) \quad \forall n \geqslant 3. \end{aligned}$$

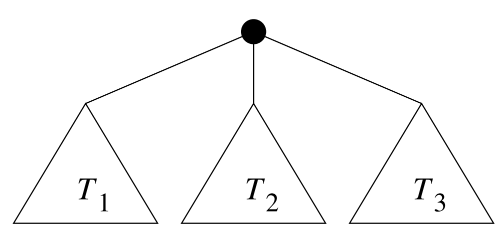
Use complete induction to prove that $f(n) = 3^n - 2^n$ for every positive integer n.

3. Recall the recursive definition of *complete binary trees*:

- a single node is a complete binary tree,
- **if T_1 and T_2 are complete binary trees *with the same height*, then the tree constructed by placing T_1 and T_2 under a new root node (as illustrated below on the left) is also a complete binary tree,**
- nothing else is a complete binary tree.



complete binary tree



complete ternary tree

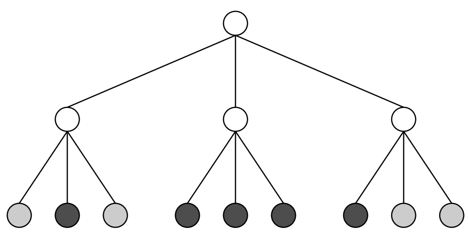
Similarly, we can define *complete* ***ternary*** *trees*:

- a single node is a complete ternary tree,
- **if T_1, T_2, and T_3 are complete ternary trees *with the same height*, then the tree constructed by placing T_1, T_2, and T_3 under a new root node (as illustrated above on the right) is also a complete ternary tree,**
- nothing else is a complete ternary tree.

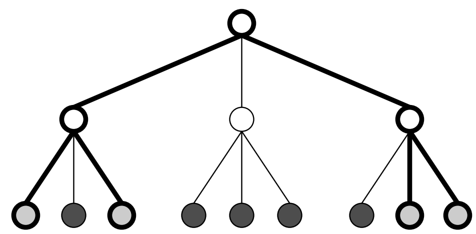
A *coloured* complete ternary tree is a complete ternary tree in which every leaf node has been assigned one of two colours (red or blue). Only the leaves are coloured — internal nodes are colourless.

A *subtree* of a tree T is a subset of T's leaves along with all of the leave's ancestors (and the edges connecting them). Every complete ternary tree contains many complete binary subtrees — simply pick two children to keep and one to remove for every internal node.

A *monochromatic* complete binary subtree of a coloured complete ternary tree is a complete binary subtree all of whose leaves are the same colour. For example, the coloured complete ternary tree on the left below contains a monochromatic complete binary subtree, indicated on the right by thick edges and node outlines.



a coloured complete ternary tree



a monochromatic complete binary subtree

Use structural induction to prove that every coloured complete ternary tree contains some monochromatic complete binary subtree.

4. Consider the following recursive definition of the set of all *propositional formulas* $\mathcal{F}$ (Definition 5.1 in Prof. Hadzilacos' notes):

   - $\forall i \in \mathbb{N}, p_i \in \mathcal{F}$ (p_i is a *propositional variable*);
   - $\forall P \in \mathcal{F}, \neg P \in \mathcal{F}$;
   - $\forall P_1, P_2 \in \mathcal{F}, (P_1 \wedge P_2) \in \mathcal{F} \wedge (P_1 \vee P_2) \in \mathcal{F} \wedge (P_1 \Rightarrow P_2) \in \mathcal{F} \wedge (P_1 \Leftrightarrow P_2) \in \mathcal{F}$;
   - $\mathcal{F}$ contains no other element.

   Recall that propositional formulas P_1 and P_2 are *logically equivalent* if P_1 and P_2 evaluate to the same value, no matter how their variables are set.

   Use structural induction to prove that for all propositional formulas $P \in \mathcal{F}$, there is an equivalent propositional formula $P' \in \mathcal{F}$ such that the only negation symbols ("$\neg$") in P' are applied to individual propositional variables.

   In your proof, you may make use of any of the equivalences listed on page 30 of the CSC 165 lecture notes — see the course website for a link to the CSC 165 notes online.