COMS20010 — Problem sheet 1

This problem sheet covers week 1, focusing on induction and O-notation. You don't have to finish the problem sheets before the class — focus on understanding the material the problem sheet is based on. If working on the problem sheet is the best way of doing that, for you, then that's what you should do, but don't be afraid to spend the time going back over the quiz and videos instead. (Then come back to the sheet when you need to revise for the exam!)

As with the Blackboard quizzes, question difficulty is denoted as follows:

- ⋆ You'll need to understand facts from the lecture notes.
- ⋆⋆ You'll need to understand and apply facts and methods from the lecture notes in unfamiliar situations.
- ⋆⋆⋆ You'll need to understand and apply facts and methods from the lecture notes and also move a bit beyond them, e.g. by seeing how to modify an algorithm.
- ⋆⋆⋆⋆ You'll need to understand and apply facts and methods from the lecture notes in a way that requires significant creativity. You should expect to spend at least 5–10 minutes thinking about the question before you see how to answer it, and maybe far longer. Only 20% of marks in the exam will be from questions set at this level.
- ⋆⋆⋆⋆⋆ These questions are harder than anything that will appear on the exam, and are intended for the strongest students to test themselves. It's worth trying them if you're interested in doing an algorithms-based project next year — whether you manage them or not, if you enjoy thinking about them then it would be a good fit.

1 Big-O Notation

1. (a) [⋆] Show that $5n \in O(n^2)$ from the definition.

   (b) [⋆⋆] Show that $2n^2 + 3n + 1 \in O(n^2 + 5n + 4)$ from the definition.

   (c) [⋆⋆⋆]

   i. Show that
   $$\sum_{i=1}^{n} i^4 \in O(n^5).$$

   ii. Show that
   $$\sum_{i=1}^{n} i^4 = \Omega(n^5).$$

   iii. Show that
   $$\sum_{i=1}^{n} i^4 = \Theta(n^5).$$

   Hint: What does Ω mean in terms of O and Θ?

   (d) [⋆⋆] Show that $3^n = 2^{O(n)}$.

   (e) [⋆⋆] Show that $n + \log n = O(n)$.

2. Why don't the following alternative definitions for O-notation (denoted by $O^\dagger$) mean the same thing as the usual definition? For each one, give an example of functions f and g such that $f(n) \in O(g(n))$ but $f(n) \notin O^\dagger(g(n))$, or such that $f(n) \in O^\dagger(g(n))$ but $f(n) \notin O(g(n))$.

(a) [★★] $f(n) \in O^\dagger(g(n))$ if there exists $n_0 > 0$ such that for all $n \geq n_0$, $f(n) \leq g(n)$.

(b) [★★] $f(n) \in O^\dagger(g(n))$ if there exists $C > 0$ such that $f(n) \leq Cg(n)$ for all $n \geq 0$.

(c) [★★★] $f(n) \in O^\dagger(g(n))$ if there exists $C > 0$ and a sequence of integers $n_0 \leq n_1 \leq n_2 \leq \dots$ such that $f(n_i) \leq Cg(n_i)$ for all i.

3. (a) [★★★] Prove that if $g(n) \in \omega(1)$ and $f(n) \in o(g(n))$, then $2^{f(n)} \in o(2^{g(n)})$. (This was used in lectures as a way of dealing with unpleasant exponential functions.)

(b) [★★] What happens if $g(n) \in o(1)$?

(c) [★★] Prove that it is **not** true that if $2^{f(n)} \in o(2^{g(n)})$ then $f(n) \in o(g(n))$.

2 Induction

4. (a) [★★] Prove by induction that for all $n \geq 1$,

$$\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}.$$

(b) [★★] Prove that for all $n \geq 1$,

$$\sum_{i=1}^{n} i \cdot i! = (n+1)! - 1$$

(c) [★★] Prove that for all $n \geq 2$,

$$\prod_{i=2}^{n} \left(1 - \frac{1}{i^2}\right) = \frac{n+1}{2n}$$

(d) [★★★] Let $x_1 = 1$, $x_{n+1} = \sqrt{1 + 2x_n}$ for all $n \geq 1$. Prove $x_n < 4$ for all $n \geq 1$.

(e) [★★★] Recall from COMS10007 that the *Fibonacci sequence* is given by $F_0 = 0$, $F_1 = 1$, and $F_{n+2} = F_{n+1} + F_n$ for all $n \geq 0$. Prove that writing $\phi = (1 + \sqrt{5})/2$ and $\psi = (1 - \sqrt{5})/2$, we have

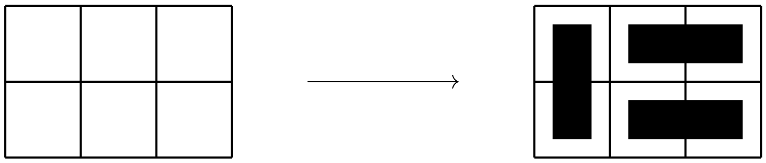
$$F_n = \frac{\phi^n - \psi^n}{\sqrt{5}} \text{ for all } n \geq 0.$$

Prove this by induction on n.

Hint: It may help to prove that $\phi^2 = \phi + 1$ and $\psi^2 = \psi + 1$.

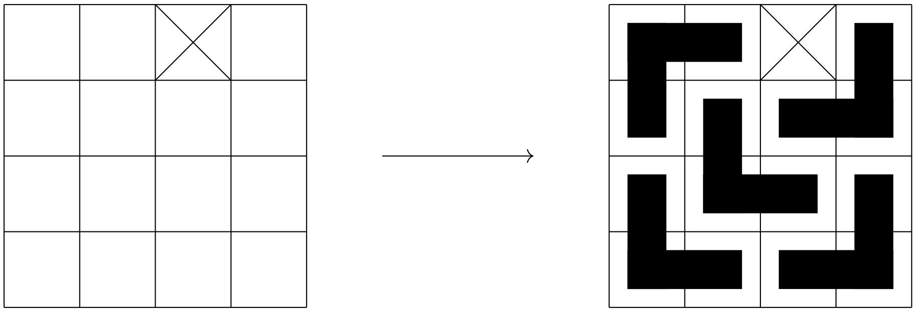
5. [★★★★] Recall from Question 4 that F_t is the t'th Fibonacci number. Prove by induction that $F_{m+n+1} = F_{m+1}F_{n+1} + F_mF_n$ for all $m, n \geq 0$.

6. [★★★] Consider a rectangular board divided into an $2 \times n$ square grid for some $n \geq 1$. We can use a "domino" to cover any two horizontally or vertically adjacent squares of the board, as shown below for $n = 3$.



Prove by induction that there are exactly F_{n+1} different ways to cover every square of the board with no two dominoes overlapping, where F_{n+1} is defined as in Question 4.

7. [★★★★★] Consider a rectangular board divided into an $2^n \times 2^n$ square grid for some $n \geq 1$, with one square missing. Instead of dominoes as in question 6, we wish to cover this board with non-overlapping corner-shaped trinominoes, as shown below for $n = 2$.



Prove by induction on n that this is always possible, no matter which square is missing.