Worth: 2% **Due:** By 8:59pm on Tuesday 27 January

Remember to write your *full name* and *student number* 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). For example, indicate clearly the **name** of every student with whom you had discussions, the **title and sections** of every textbook you consulted (including the course textbook), the **source** of every web document you used (including documents from the course webpage), 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.*

A new "healthy fast-food" restaurant sells *Tofu Nuggets* in boxes that contain either 3, 10 or 25 nuggets. You visit the restaurant with a group of your friends. Can you purchase exactly the number of nuggets required so that everyone gets to try one, but nobody has to try more than one?

We formalize the "Tofu Nugget" problem as follows.

- **Input:** An integer $N \geqslant 0$ (the size of your group).
- **Question:** Determine whether or not you can purchase exactly N Tofu Nuggets (given that you can only purchase boxes of 3, 10 or 25).

Even though there is nothing to optimize in this problem, it is possible to solve it with Dynamic Programming.

(a) Describe the recursive structure of solutions to this problem. Be clear and precise: describe the manner in which a solution for the problem relates to solutions for sub-problems, what those sub-problems are exactly, and why the relationship is as you described it.

(b) Define an array suitable for solving the Tofu Nugget problem through dynamic programming. State clearly what value is stored in each array location, the relationship between the array index/indices and the sub-problem being solved, and the range of valid indices.

Keep in mind the important distinction discussed in class between the *array definition* and the *recurrence relation*.

(c) Give a recurrence relation for your array from part (b). Remember to include base case(s). Justify your recurrence from the structure of the problem described in part (a).

Keep in mind the important distinction discussed in class between the *recurrence relation* and the *algorithm*.

(d) Write an iterative algorithm that solves the Tofu Nugget problem, based on your recurrence relation from part (c). Then, give a tight bound on the worst-case complexity of your algorithm. Does your algorithm run in polynomial time? Explain.