Department of Computing and Information Systems
COMP90038 Algorithms and Complexity Tutorial Week 4

The exercises

1. **function** SECRET($A[0..n-1]$)
    // Input: An array $A[0..n-1]$ of n real numbers
    // Output: ?
    $minval \leftarrow A[0]$; $maxval \leftarrow A[0]$
    for $i \leftarrow 0$ to $n-1$ **do**
        if $A[i] < minval$ **then**
            $minval \leftarrow A[i]$
        if $A[i] > maxval$ **then**
            $maxval \leftarrow A[i]$
    return $maxval - minval$

    (a) What does this algorithm compute?
    (b) What is its basic operation?
    (c) How many times is the basic operation executed?
    (d) What is the time complexity of the algorithm (in a Big-O sense)?

2. One possible way of representing a polynomial

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

    is as an array A of length $n+1$, with $A[i]$ holding the coefficient a_i.

    (a) Design a brute-force algorithm for computing the value of $p(x)$ at a given point x. Express this as a function PEVAL(A, n, x) where A is the array of coefficients, n is the degree of the polynomial, and x is the point for which we want the value of p.
    (b) If your algorithm is $\Theta(n^2)$, try to find a linear algorithm.
    (c) Is it possible to find an algorithm that solves the problem in sub-linear time?

3. Trace the brute-force string search algorithm on the following input: The path p is '`needle`', and the text t is '`there_need_not_be_any`'. How many comparisons (successful and unsuccessful) are made?

4. Assume we have a text consisting of one million zeros. For each of these patterns, determine how many character comparisons the brute-force string matching algorithm will make:

    (a) `010001`   (b) `000101`   (c) `011101`

5. Give an example of a text of length n and a pattern, which together constitute a worst-case scenario for the brute-force string matching algorithm. How many character comparisons, as a function of n, will be made for the worst-case example.

6. The *assignment problem* asks how to best assign n jobs to n contractors who have put in bids for each job. An instance of this problem is an $n \times n$ *cost matrix* C, with $C[i, j]$ specifying what it will cost to have contractor i do job j. The aim is to minimise the total cost. More formally, we want to find a permutation $\langle j_1, j_2, \ldots j_n \rangle$ of $\langle 1, 2, \ldots, n \rangle$ such that $\sum_{i=1}^{n} C[i, j_i]$ is minimized.

   Use brute force to solve the following instance:

	Job 1	**Job 2**	**Job 3**	**Job 4**
Contractor 1	9	2	7	8
Contractor 2	6	4	3	7
Contractor 3	5	8	1	8
Contractor 4	7	6	9	4

7. Give an instance of the assignment problem which does not include the smallest item $C[i, j]$ of its cost matrix.

8. Outline an exhaustive-search algorithm for the Hamiltonian circuit problem.