Data Structures and Algorithms

Computer Science Degree, Group I

First Semester Exam – February 12, 2015.

1. (3.5 points) For a given vector with n elements ($n \geq 0$), we need to code an iterative algorithm that calculates the sum of the products of all the pairs of numbers contained in different positions of the vector (with a complexity in $\mathcal{O}(n)$).

Examples: for the vector $[1, 3, 5, 7]$ it must return $1*3+1*5+1*7+3*5+3*7+5*7$.

For the vector $[6, 2, 5, 9, 1, 2]$ it must return

$6*2+6*5+6*9+6*1+6*2+2*5+2*9+2*1+2*2+5*9+5*1+5*2+9*1+9*2+1*2$

You are asked to:

1. Specify the algorithm.
2. Derive and code (or code and verify) the algorithm.
3. Calculate and justify the complexity of the algorithm.

2. (3 points) The *specular image* of a natural number is the number that results when we invert its digits. For example, the *specular image* of 13492 is 29431 and the *specular image* of 1000 is 1. Code two recursive algorithms, one tail recursive and one non tail recursive, that calculate the *specular image* of a natural number represented as an `unsigned int`. Write the initial invocation to both algorithms and calculate their complexity.

NOTE: You do not have to specify, design or verify the algorithms.

3. (3.5 points) We have a list of n dietary products that we have been provided in order to design a balanced diet. For each product i ($0 \leq i < n$) we know its price $p_i \geq 0$, its protein content $q_i \geq 0$ and its caloric load $c_i \geq 0$. We need to select some of these products (not more than one unit of each of them) so that the total price does not exceed a certain budget M, the protein content is at least Q and the caloric load is as low as possible. Implement a backtracking algorithm that finds the optimal selection, i.e. the one that minimizes the caloric load. Appropriate pruning of the search tree will be a plus.