CSC 165 H1S **Homework Assignment # 3** **Winter 2009**

Worth: 8% **Due:** By 6pm on Wednesday 1 April

1. Consider the Selection Sort algorithm below, that sorts an input array A into non-decreasing order.

```
SS(A):
    i = 0
    while i < len(A):
        s = i
        j = i + 1
        while j < len(A):
            if A[j] < A[s]:
                s = j
            j = j + 1
        t = A[i]
        A[i] = A[s]
        A[s] = t
        i = i + 1
```

For each question below, find the *exact* answer **without** using $\mathcal{O}$, Ω, or Θ—count one step for each statement in the algorithm.

(a) Compute the exact number of steps performed by SS on input $[4, 2, 1, 3, 5]$.

(b) Compute the exact worst-case number of steps performed by SS on any input of size 3.

(c) For all $n \in \mathbb{N}$, compute the exact worst-case number of steps performed by SS on any input of size n.

2. Prove that $T_{\text{BFT}}(n) \in \Theta(n^2)$, where BFT is the algorithm below.

```
BFT(E, n):
    i = n - 1
    while i > 0:
        P[i] = -1
        Q[i] = -1
        i = i - 1
    P[0] = n
    Q[0] = 0
    t = 0
    h = 0
    while h <= t:
        i = 0
        while i < n:
            if E[Q[h]][i] != 0 and P[i] < 0:
                P[i] = Q[h]
                t = t + 1
                Q[t] = i
            i = i + 1
        h = h + 1
```

(Although this is not directly relevant to the question, this algorithm carries out a breadth-first traversal of the graph on n vertices whose adjacency matrix is stored in E.)

3. Prove or disprove that for all functions $f, g : \mathbb{N} \to \mathbb{R}^+$, $g \in \mathcal{O}(f) \Rightarrow g^2 \in \mathcal{O}(f^2)$ (where the function f^2 is defined as $f^2(n) = f(n) \cdot f(n)$ for all $n \in \mathbb{N}$, and similarly for g^2). Write a detailed structured proof of your claim.

4. Prove or disprove that for all functions $f, g : \mathbb{N} \to \mathbb{R}^+$, $g \in \Omega(f) \Rightarrow \lfloor g \rfloor \in \Omega(f)$ (where the function $\lfloor g \rfloor$ is defined as $\lfloor g \rfloor(n) = \lfloor g(n) \rfloor$ for all $n \in \mathbb{N}$). Write a detailed structured proof of your claim.

5. How many non-zero numbers are there in the normalized floating-point system with $\beta = 4$, $t = 4$, $e_{min} = -4$, $e_{max} = 4$? Show your work.

6. Does the multiplication of numbers exactly representable in a floating-point system always produce numbers exactly representable in the system? Justify.

7. In lecture, we calculated an upper bound of $\beta^{1-t}/2$ for the relative error introduced by rounding to nearest in a floating-point system. Compute an upper bound for "round towards zero" (*i.e.*, round down for positive numbers, round up for negative numbers).