CS 4349 — Advanced Algorithm Design and Analysis, Homework 1

https://iniyanjoseph.com | iniyanijoseph@gmail.com

Question 1

a

```
1  for each i = 1, ..., n − 1 do
2      for each j = 1, ..., n − i
3          A′ ← A
4          if A[j] > A[j + 1] do
5              A′[j + 1] ← A[j]
6              A′[j] ← A[j + 1]
7          end
8          A ← A′
9      end
10 end
```

b

After the k-th iteration of the outer loop, the k-th largest element of the array will at be the n-i+1th index. For example, after the first iteration of the outer loop, the largest element of the array will be at the nth index of the array.

c

The outer loop uses $n-1$ loops for an input array of length n. The last loop is redundant, as it is guaranteed that the smallest index will be located at index 1 and indices 2...n will already be sorted. Therefore, no swaps will be performed. This is additional comparison is asymptotically negligible however.

Question 2

$F_k = F_{k-1} + F_{k-2}$ where $F_1 = F_2 = 1$ We must show that $\sum_{i=1}^{n} F_i^2 = F_n * F_{n+1}$ **Proof.** Basis : We must show that $\sum_{i=1}^{2} F_i^2 = F_2 * F_3$.

$$F_1^2 + F_2^2 = 1^2 + 1^2 = 2$$
$$F_2 * F_3 = 1 * 2 = 2$$

Inductive Step:
We assume that $\sum_{i=1}^{n} F_i^2 = F_n * F_{n+1}$ as our inductive hypothesis.
We must show that $\sum_{i=1}^{n} F_i^2 + F_{n+1}^2 = F_{n+1} * F_{n+2}$

$$\sum_{i=1}^{n} F_i^2 + F_{n+1}^2$$
$$= F_n * F_{n+1} + F_{n+1}^2$$
$$= F_{n+1} * (F_n + F_{n+1})$$
$$= F_{n+1} * (F_{n+2})$$

Question 3

There is no greatest negative rational number **Proof.** Assume for contradiction that there is a greatest rational number. This then may be denoted as $-\frac{a}{b}$, where $a, b \in \mathbb{N}$.
Take another number $-\frac{a}{b+1}$. $|\frac{a}{b+1}| < |\frac{a}{b}|$, and therefore $-\frac{a}{b+1}$ is greater than $-\frac{a}{b}$. This is a contradiction, thus there is no greatest rational number.

Question 4

a

Does $o \Rightarrow O$ and $\omega \Rightarrow \Omega$ both hold for two functions?
Yes. $o \Rightarrow O$ since $o(f(n)) \subset O(f(n))$ and $\omega(f(n)) \subset \Omega(f(n))$

b

Prove by counterexample that $O \Rightarrow \Theta$ is false.
Assume $O \Rightarrow \Theta$. Informally, this means that if a function is upper bounded by a function f, then it must also be lower bounded by f. For counterexample, take n^2. $n^2 = O(n^3)$. However, $n^2 \neq \Omega(n^3) \Rightarrow n^2 \neq \Theta(n^3)$.
Thus, $O \nRightarrow \Theta$

c

Prove directly that $f(n) = \Omega(f(n))$
This follows from the definition of $\Omega(f(n))$.
A function $g(n)$ is $\Omega(f(n)) \Leftrightarrow g(n) \geq c * f(n)$ for all $n \geq n_0$
Since $f(n) = f(n)$, it is trivially true that $f(n) = 1 * g(n)$ for all $n \geq 1$.

d

Prove by counterexample that $f(n) = O(f(n))$
A function $g(n)$ is $O(f(n)) \Leftrightarrow g(n) \leq c * f(n)$ for all $n \geq n_0$.
Note that $f, g : \mathbb{N}_{>0} \rightarrow \mathbb{N}_{>0}$
For counterexample, assume there is exists a c, n_0 for which $f(n) \leq c * f(n)$ for some $n \geq 0$.
This means that $f(n) * 2 < f(n)$ for n $> n_0$, however this is a contradiction since $f(n)$ must be strictly greater than 0, and $ka \geq a$ for $k, a \in \mathbb{N}_{>0}$

Question 5

In the first iteration of the loop, $n - 1$ comparisons (and swaps) will be required to move the smallest element of the array to the nth index of the array. In the second, the previously sorted elements need not be compared, so $n - 2$ swaps will be necessary.
This pattern continues until only 1 swap is necessary.
$\sum_{i=1}^{k} k - i = \frac{(k)(k-1)}{2}$.
For insertion sort, this means that $\frac{(n-1)(n-2)}{2}$ comparisons will be necessary.