CS 101, Assignment 1

Christopher Huynh, Oliver Rene

April 2016

1 Question 1

1.1

Let n = A.length
For i = 1 to n:
    For j = 0 to n-1:
        if A[j] >= A[j+1]:
            Swap A[j] and A[j+1]

1.2

θ (n^2)=(n-1)^2
With C_1=1 and C_2=$\frac{1}{4}$ and n_0 = 2

1.3

Sub array $A[[n-i]..n]$ is sorted and any element in $A[[1]..[i]]$
For base case i=1, algorithm does nothing. Array of size 1 is in sorted order.
Given the sub array $A[[n-i]..[n]]$ i inserts the largest of the unsorted elements of $A[[1]..[i]]$ as computed by the j loop. $A[[1]..[i]]$ contains only elements smaller than $A[[n-i]..[n]]$ and $A[n]$ is greater than any element in $A[[1]..[n-1]]$ is sorted and the invariant is preserved.
At the last iteration $A[[1]..[n]]$ is sorted.

2 Question 2

Hypothesis: Let n $\in$ N. Let there 2^{n-1} subsets of A of an odd size. Let P(n) be the statement that any set with $\{1, 2, ..., n\}$ elements has 2^{n-1}.
Base case: n=1. set is 1. There should be $2^{(1)-1}$=1 subsets. There is only one subset of 1 which with an odd number of elements which is 1
Induction: Let k $\in$ N. Now the array goes from $\{1, 2, ..., k, k+1\}$ has $2^{(k+1)-1}$ = 2^k subsets. Dropping the k+1 term from the set results in a subset A that has 2^{k+1} subsets once more. Set B has k elements in it, which is exactly the

set A without its k+1 element, therefore by induction hypothesis there are 2^{k-1} subsets of odd elements.
If we combine type I with type II to make all subsets of A there are $2^(k-1) + 2^(k-1) = 2^(k)$, thus completing the inductive step.

3 Question 3

For the upper bound of f(n) we simply to choose a c_1 which satisfies the condition that $\lim_{k\to\infty} \frac{a_k n^k}{c_1 n^k} = \frac{a_k}{c_1} < 1$. For the lower we similarly need to compute a c_2 that satisfies $\lim_{k\to\infty} \frac{a_k n^k}{c_2 n^k} = \frac{a_k}{c_2} > 1$. Solving algebraical $a_k < c_1$ and $a_k > c_2$. As you take $\lim_{k\to\infty} \frac{n^{k+1}}{n^k}$ O(n) grows to infinity, whereas f(n) will be unable to catch up to it, growing to infinity, albeit at a slower rate.

4 Question 4

Choosing c=1 n_0=16, there exists a c and n_0 such that $0 <= \log_2 n <= c * \sqrt{n}$ for all n $>= cn_0$
$\lim_{k\to\infty} \frac{\log_2 n}{\sqrt{n}} = 0$ which is less than any constant c that makes sense. Therefore it contradicts the definition of an Ω time complexity.