Q1.

```
for i=1 to n-1:
    for p=1 to n-k:
        if (A[p] < A[p-1])
            swap A[p] and A[p-1]
```

The outer loop of bubble sort will run n times regardless of the inner loop. The inner loop will run from 1 to $n-i$ times, so it'll run $n-i$, $n-i-1$ ... times subtracting by 1 every time it runs. This is just $\frac{cn^2}{2} - \frac{cn}{2}$ which is just $\frac{cn}{2}$ times. The outer loop multiplied by the inner loop will be $\frac{cn}{2} \cdot n$ which is $\frac{cn^2}{2}$ which is just $\Theta(n^2)$

Invariant: After the outer loop runs k times the last n-k elements are sorted and the last n-k elements are larger than to the left of n-k.

Base Case (n=1): The last n-1 elements is sorted since the inner loop will bring the largest element in the array to n-1, which is the last element.

Proof by induction (k+1): After k runs, the last n-k elements will be sorted by the invariant. If we add 1 more run to the array, the largest element from A[1] to A[n-k] will be moved to A[n-k-1]. Everything from A[1] to A[n-k-2] will then be less than A[n-k-1] to A[n] and thus, proves the invariant.

Q2. Base Case ($n=1$): $\{1\}$ $\quad 2^{1-1} = 2^0$

1 odd subset in $\{1\} = 1$

Hypothesis: The number of subsets in $\{1, 2, \ldots, n\}$ will have an odd number of elements is 2^{n-1}

Inductive Step: Assume $2^{k-1} = \{1, 2, \ldots, k\}$ of having an odd # of subsets is true.

Prove that $2^{k-1+1} = \{1, 2, \ldots, k, k+1\}$ of having an odd number of subsets is true.

Since there are 2^{k-1} even subsets in $\{1, 2, \ldots, k\}$ by the definition of the Cardinality. If we append $k+1$ to the end of every subset than we will get 2^k odd subset.

$$\underbrace{2^k}_{\text{total Subset}} - \underbrace{2^{k-1}}_{\text{odd Subsets}} = \underbrace{2^{k-1}}_{\text{even subsets}}$$

Appending $k+1$ to the end of every even subset will yield us

$$2^{k-1} + 2^{k-1} = 2^k$$
$$= 2^{k+1-1}$$

$\boxed{\square}$

Q3. $f(n) = a_0 + a_1 \cdot n + a_2 \cdot n^2 + \dots + a_k n^k$

Show $f(n) \in \Theta(n^k)$

By the definition of big theta
$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$ where $c \neq 0$ then $f(n) = \Theta(g(n))$

plugging in $f(n)$ and $g(n)$ we get

$$\lim_{n \to \infty} \frac{a_0 + a_1 n + a_2 n^2 + \dots + a_k n^k}{n^k}$$

if we divide everything by n^k everything will go to $\emptyset$ but the denominator and the last element

$$\lim_{n \to \infty} \frac{\frac{a_0}{n^k} + \frac{a_1 n}{n^k} + \frac{a_2 n^2}{n^k} + \dots + \frac{a_k n^k}{n^k}}{n^k / n^k} = \frac{0 + 0 + 0 + \dots + a_k}{1}$$

$= a_k$ where a_k is a constant thus proving

$f(n) \in \Theta(n^k)$

Prove $f(n) \notin O(n^{k'})$ for all $k' < k$

By definition of Big O $\lim_{n \to \infty} \frac{f(n)}{g(n)} < \infty$ then $f(n) = O(g(n))$

plugging in $f(n)$ and $g(n)$ we get

$\lim_{n \to \infty} \frac{a_0 + a_1 n + a_2 n^2 + \dots + a_k n^k}{n^{k'}}$ since k' is less than k

$$\lim_{n \to \infty} \frac{\frac{a_0}{n^{k'}} + \frac{a_1 n}{n^{k'}} + \frac{a_2 n^2}{n^{k'}} + \dots + \frac{a_{k'} n^{k'}}{n^{k'}} + \dots + \frac{a_k n^k}{n^{k'}}}{n^{k'} / n^{k'}}$$

Since the limit of n goes to infinity we get

$$\frac{0+0+0+\ldots+a_{k'}+\ldots+\infty}{1}$$

and thus $\lim_{n\to\infty} \frac{f(n)}{g(n)} \notin O(g(n))$ by contridiction.

∎

Q4. Prove $\log_2 n = O(n^{1/3})$, but $\log_2 n$ is not $\Omega(n^{1/3})$. Is $\log_2 n = \Theta(n^{1/3})$? why or why not?

By the definition of Big O and Big Ω we have

If $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$ $f(n) = O(g(n))$ and

If $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$ $f(n) = \Omega(g(n))$

plogging in $f(n)$ and $g(n)$ we get

$\lim_{n\to\infty} \frac{\log_2 n}{n^{1/3}}$ by L'Hospital's rule we will get

$$\lim_{n\to\infty} \frac{\frac{1}{n\ln(2)}}{\frac{1}{3}n^{-2/3}} = \lim_{n\to\infty} \frac{\frac{1}{n\ln(2)}}{\frac{1}{3n^{2/3}}} = \lim_{n\to\infty} \frac{1}{n\ln(2)} \cdot \frac{3n^{2/3}}{1}$$

$$\lim_{n\to\infty} \frac{3n^{2/3}}{n\ln 2} = \lim_{n\to\infty} \frac{3n^{2/3}/n}{n\ln 2/n}$$

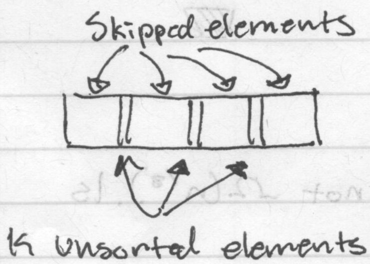
applying limits we get

$$\frac{0}{\ln 2} = 0$$

Thus we get $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ which is less than ∞ so we get $O(n^{1/3})$ but is not greater than 0 so $\log_2 n \notin \Omega(n^{1/3})$ Since $\log_2 n$ is $O(n^{1/3})$ but not $\Omega(n^{1/3})$ it is not $\Theta(n^{1/3})$

∎

Q5. Since there are already $n-k$ that are sorted it will pass straight through those elements by the while loop, (it will only make 1 comparison). This means that it will make k comparisons in the inner loop. Since the outer loop makes n comparisons regardless of sorted elements it is just $n \cdot k$.



$O(n)$ outer loop

$O(k)$ innerloop

$O(n \cdot k) = O(nk)$