Asymptotic Complexity

Q1. What does the following mystery function do? What is its asymptotic time complexity?

```
// A is an unsorted array of integers
int mystery(A[], int start, int end) {
  if (start == end)
    return A[start];
  else {
    mid = (start + end)/2;
    var1 = mystery(A, start, mid);
    var2 = mystery(A, mid+1, end);
    return min(var1,var2);
  }
}
```

This was discussed in class. mystery() returns the smallest element in array A. Time complexity T(n) is in O(n).

Q2. If f(n) is O(g(n)) and g(n) is $O(n^3)$ then f(n) is $O(n^3)$

TRUE / FALSE (explain your answer)

TRUE.
By transitivity, if f(n) is O(g(n)) and g(n) is O(h(n)) then f(n) is O(h(n))

Q3. If f(n) is $O(n^2)$ and g(n) is $O(n^2)$ then f(n) is O(g(n))

TRUE / FALSE (explain your answer)

FALSE.
Suppose g(n) = c n + b and f(n) = c n^2 + b n + a, where a, b and c are constants.
f(n) is $O(n^2)$ and g(n) is $O(n^2)$ but f(n) is **not** O(g(n))