1. Approximately how many array accesses does the following code fragment make as a function of $N$? (Assume the compiler does not optimize away any array accesses in the innermost loop.)

```c
int sum = 0;
for (int i = 0; i < N; i++)
    for (int j = i + 1; j < N; j++)
        for (int k = 1; k < N; k = k*2)
            if (a[i] + a[j] >= a[k]) sum++;
```

2. List the following functions from highest to lowest order. If any are of the same order, circle them together in your list.

$$2^n \quad \lg \lg n \quad n^3 + \lg n \quad \lg n \quad n - n^2 + 5n^3 \quad 2^{n-1} \quad n^2 \quad n^3$$

$$n \lg n \quad (\lg n)^2 \quad \sqrt{n} \quad 6 \quad n! \quad n \quad (3/2)^n$$

Questions 3 and 4 on reverse side →
3. For each of the following pairs of functions $f(n)$ and $g(n)$, either $f(n)=O(g(n))$ or $g(n)=O(f(n))$, but not both. Determine which is the case (you do not have to show a proof).

a. $f(n) = \frac{n^2 - n}{2}$, $g(n) = 6n$

b. $f(n) = n + 2\sqrt{n}$, $g(n) = n^2$

c. $f(n) = n + n \log n$, $g(n) = n\sqrt{n}$

d. $f(n) = n^2 + 3n + 4$, $g(n) = n^3$

e. $f(n) = n \log n$, $g(n) = n\sqrt{n} / 2$

f. $f(n) = n + \log n$, $g(n) = \sqrt{n}$

g. $f(n) = 2 (\log n)^2$, $g(n) = \log n + 1$

h. $f(n) = 4n \log n + n$, $g(n) = (n^2 - n) / 2$

4. Is $2^{n+1} = O(2^n)$? Show a proof of your answer.