1.

with a1=3, a2=5:

a3 = 5+2\*3-2 = 9 = 23+1

a4 = 9+2\*5-2 = 17 = 24+1

Assume for all n (k >= n >= 2): an=2n+1. Now, we need to prove: an+1=2n+1+1

an+1 = an + 2an-1 – 2 = 2n+1 + 2(2n-1+1) – 2 = 2n+1 + 2n+ 2 – 2 = 2n+1+1

So, we proved with all n>=2: an=2n+1

2.

T(n) = 2T(n/2) + n = 2(2T(n/4) + n/2) + n = 4T(n/4) + 2n = 4(2T(n/8) + n/4) + 2n = 8T(n/8) + 3n

= 2KT(n/2K) + Kn

If n/2K = 2 => n=2K+1 => K = lg(n) - 1

T(n) = n/2T(2) + n(lg(n) – 1) = n/2\*2 + nlg(n) – n = nlg(n)

3.

T(n) = 3T(n/2) + n

The node depth at i is n/2i

So, the tree has lg(n)-1 levels and 1 leaf

The total cost for all nodes at depth i with i=0,1,2...lg(n)-1: n/2i

T(n) = sumi=0->lg(n)-1(n/2i) + O(1) < sumi=0->∞(n/2i) + O(1) = 1/(1 – 1/2)\*n+ O(1) = O(n)

Guess T(n) <= cn

=> T(n) <= cn/2 + n = n(c/2+1) <= cn

=> c >= 2

4. f(n)=O(g(n)) only if |f(n)| <= C|g(n)| for every big enough n and the constant C  
a.

In case f(n) = n and g(n) = n2

Then f(g(n)) = n2 different from O(f(n))=O(n)

So, the answer is NO

b.

Assume f(n) = n, g(n) = 1

f(n)+g(n) = n+1 different from Θ(1)

c.

Assume this statement is correct, then there is a constant C’ for which f(n) <= C’f(n/2)f(n) = Ω(f(n/2))

=> There is a constant C for which f(n) >= Cf(n/2)

Assume f(n) = 2n  then 2n > C\*2n/2

So the statement is incorrect

d.

Assume n >= A (A∈N)

There is a constant C such that C>0 and f(n) <= Cg(n)

Let K = 1/C, we have: g(n) <= Kf(n)

For all n >= A, K > 0, this prove g(n) is Ω(f(n))

e.

Assume f(n) = 1/n => f(n)/f(n)2 = n , unbounded

Therefor f(n) ≠ O((f (n))2)

5.

a. f(n) = n3 + 15n + 2

g(n) = n4

We need to prove

f(n) <= Cn4 for n>=n0 . Then:

1/n + 15/n3 + 2/n4 <= C

=> choose n0 = 1 and C=18

f(n) / g(n) = (n3 + 15n + 2)/n4 <= 18

b. f(n) = 2n3 + 25n

g(n) = n2

Assume 2n3 + 25n >= Cn2 then:

2n + 25/n >= C

The minimum value of the left side with n = sqrt(12.5)

=> Therefore, n >= n0 = 4 and C = 14