For the following functions, give the best Big O( ) descriptions:

```
def make_list2(n):
    res = []
    for i in range(n):
        val = i**3 - 21
        res.insert(0, val)
    return res
```

Answer: $O(n^2)$

```
def do_stuff2(n, x=1.23):
    if n <= 0:
        return 0
    val = 1
    for i in range(n//2):
        for j in range(n//4):
            x += 2*x + j/2 + i*1.2
    while val <= n:
        for i in range(n):
            x += val**2 + i//2
        val *= 2
    x += do_stuff2(n - 1, x/2)
    return x
```

Answer: $O(n^3)$

```
def do_thing2(n):
    x = 3.25;
    val = 1
    while val <= n:
        for j in range(n):
            x += 2*x + j/2 + val*1.2
        val *= 2
    return x
```

Answer: O(n*log(n))

```
def do_something2(n, x=0):
    if n <= 0:
        return 0
    for i in range(n):
        x += 2*x + i*1.2
    x += do_something(n//2, x - 1)
    return x
```

Answer: O(n)