Determine an exact operation count, as a function of $N$, for the execution of each of the following. Do a worst-case analysis, and determine the big-O complexity for each. Count assignments, increments, pointer dereferences and array references as single operations.

1. The double-linked list `insert()` function given on slide 4.8 of the course notes. Assume there are $N$ items in the list when the function is called.

2. The following code, where $N$ is a constant:

   ```
   int Buffer[N] = {0};
   int i, j;
   i = 1;
   while (i < N) {
       j = 0;
       while (Buffer[j] < i) {
           j++;
           Buffer[j] = i;
           i = i + 2;
       }
   }
   ```

3. The following code, where $N$ is a constant:

   ```
   int Buffer[N] = {0};
   int i, j;
   i = 1;
   while (i < N) {
       j = 0;
       while (Buffer[j] < i) {
           j++;
           Buffer[j] = i;
           i = 2 * i;
       }
   }
   ```

Consider the following recursive function:

```
int Puzzle(int base, int limit) {
    if (base > limit)
        return -1;
    else if (base == limit)
        return 1;
    else
        return (base * Puzzle(base + 1, limit));
}
```

4. What are the base cases for this function?

5. Trace the execution of the call: `int puzzler = Puzzle(4, 6);`

6. Provide a nonrecursive implementation of `Puzzle()`.