1. For this problem, assume that you have access to a procedure `Merge(A, B, C, m, n)`, that merges two sorted arrays `A[1..m]` and `B[1..n]` into a single sorted array `C[1..m+n]`. This procedure runs in `m + n - 1` comparisons. Assume we also have access to unlimited dynamic array allocation.

Suppose that you have a collection of `n`, 1-dimensional arrays each of length `n`, `D[1..n][1..n]`, such that each array is sorted (i.e. `D[i][j] <= D[i][j+1]`). We want to compute a single sorted array that contains all `n^2` elements.

The first strategy for accomplishing this is called *cascading merging*. First merge `D[1]` with `D[2]` to form an array of length `2n`, then merge this array with `D[3]` to form an array of length `3n`, etc. The final merge is between an array of length `(n - 1)n` with `D[n]` to form the final sorted array of length `n^2`.

The second strategy is called *balanced merging*. Assume that `n` is a power of 2. First merge pairs `D[1]` and `D[2]` together, `D[3]` and `D[4]` together, and so on until merging `D[n-1]` and `D[n]` together. The result is `n/2` sorted arrays of size `2n` each. Next, repeat the balanced merging on these arrays, resulting in `n/4` arrays of size `4n` each. This is repeated until there is one array of size `n^2`.

(a) What is the exact number of comparisons for cascading merging? Justify your answer. Give as simple a function `g(n)` such that the number of comparisons is `Θ(g(n))`.

(b) What is the exact number of comparisons for balanced merging? Justify your answer. Give as simple a function `g(n)` such that the number of comparisons is `Θ(g(n))`.

(c) Which algorithm is better, cascading merge or balanced merge (or is neither better)?

2. Assume merging two lists of sizes `m \leq n` takes exactly `\lceil m \lg(4n/m) \rceil` comparisons. NOTE: `n! \approx (n/e)^n`.

(a) What is the number of comparisons for cascading merging? Just get the high order term exact. Justify your answer. Give as simple a function `g(n)` as possible, such that the number of comparisons is `Θ(g(n))`.

(b) What is the number of comparisons for balanced merging? Just get the high order term exact. Justify your answer. Give as simple a function `g(n)` as possible, such that the number of comparisons is `Θ(g(n))`. 