Estimating g(x) Given f(x)

* 1. n
  2. n^2
  3. n^5
  4. n^3
  5. n^2log(n)
  6. n log(n)

Counting Operations to Produce Polynomials

* 1. f(x) = 1
  2. f(x) = n^2
  3. f(x) = n
  4. f(x) = n

More Advanced Practice

1. g(x) = log(x) for binary search because binary search is essentially splitting the sorted array in half every time you compare, thus the complexity is essentially equivalent to finding the number of 2s multiplied that can fit into the size of the array (as you're splitting the array into halves until the term is found), therefore g(x) = log2(x) or log(x) as short term
2. g(x) = n for best case scenario, g(x) = n^2 for worst case. In best case, you pass through the array once, swapping each term once, thus g(x) = n. Worst case, for each term you pass through the array once in order to swap it to its correct position, therefore g(x) = n^2