CSCI 3104 (Spring 2016): Quiz-1. **Time Limit**: 25 minutes. **Maximum points**: 20.

Your Name: Solution

P1 (10 points) Assume binary search on a sorted array of length n has worst-case running time $C_1(\log_2(n))$ for constant C_1, and merge sort has worst-case running time $C_2 n \log(n)$.

We are given an *unsorted array* a of size n and asked to search for the element e in the array. We adopt the following algorithm:

1. Divide array a it into k equal parts of size $\frac{n}{k}$. Assume k divides n.
2. For each part from 0 to $k-1$,
   (a) Sort using mergesort.
   (b) Search for e using binary search.
   (c) If e is found, return TRUE.
3. Since e is not found in any of the parts, return FALSE.

The Python code is shown below:

```
def search(array a, int k):
  n = length(a)
  p = n/k # Assume p is an integer
  for i in range(0,k):
      # PART i of array is from a[i * p : (i+1) * p]
      b = mergeSort(a[i* p : (i+1) * p ] )
      result = binarySearch(b , e)
      if (result == True)
         return True # found element e
  end
  return False # did not find e
```

1 pt As a function of n, k, what is the total number of calls to the `mergeSort`?

Two solutions are possible: k OR if the student counts the number of recursive calls, then $k(2\frac{n}{k}-1)$

1 pts As a function of n, k, what is the size of the array for each call to the `mergeSort` function? $\boxed{\frac{n}{k}}$

1 pts As a function of n, k, what is the total number of calls to the `binarySearch` ? $\boxed{k}$

7 pts Write an expression for the total running time of the algorithm in terms of n, k.

$$k(C_2\frac{n}{k}\log(\frac{n}{k}) + C_1\log(\frac{n}{k})) = (C_2 n + C_1 k)(\log n - \log k)$$

Award generous partial credit depending on how close they are. No explanations are needed.

P2 (10 points) Consider the five functions below:

$$f_1 : \log(n),\ f_2 : n,\ f_3 : n^{1.5},\ f_4 : n^2,\ f_5 : 2^n$$

For each of the functions $g(n)$ below, select the tightest possible upper bound using one of the functions f_i above, such that $g(n) \in O(f_i)$.

If no such function exists, write NONE as your answer.

(A) $g(n) = 2n^2 + 3.5n^{1.2} + \sqrt{n} + 100.$

$\boxed{f_4}$

(B) $g(n) = 3^{\log_2(n)}$. Note $\log_2(3) \sim 1.585$.

$\boxed{f_4}$

(C) $g(n) = 2^{1.5n+10}$. Note $2^{1.5} \sim 2.828$ and $2^{10} = 1024$.

$\boxed{\text{NONE}}$

(D) $g(n) = \log(n) + n^2 + 1.5 \times n^{10} + 3.5 \times 2^n$.

$\boxed{f_5}$

(E) $g(n) = n^{0.1}\log(n)$.

$\boxed{f_2}$