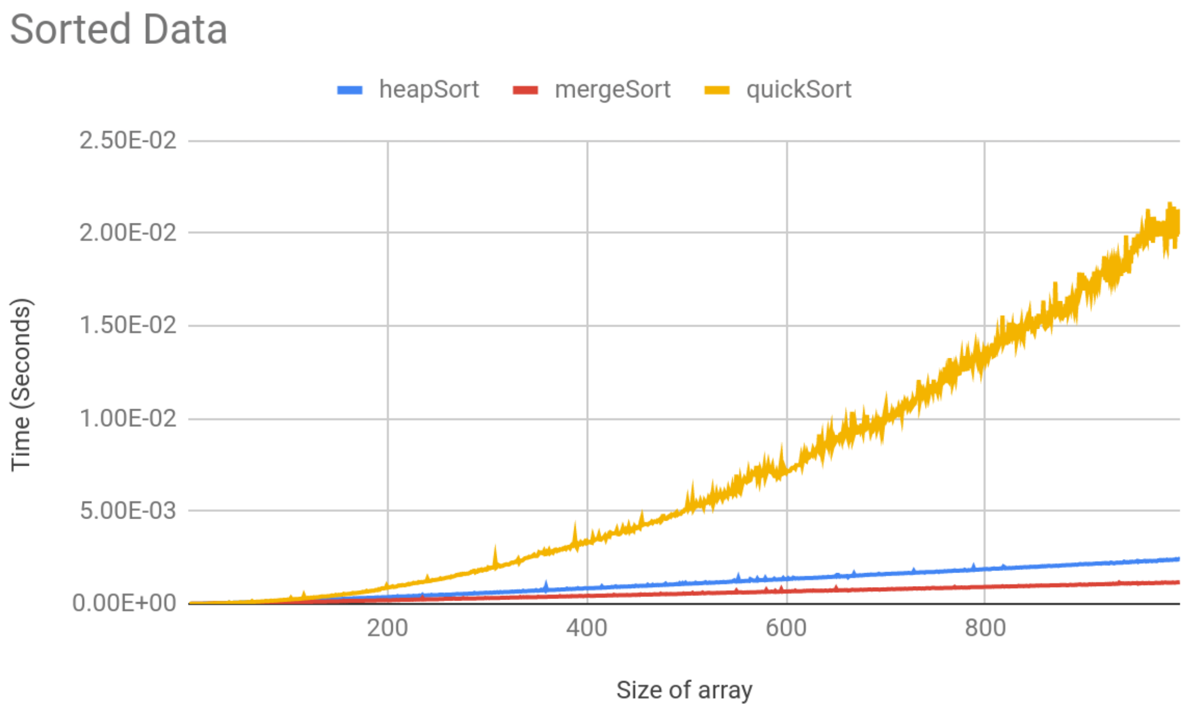
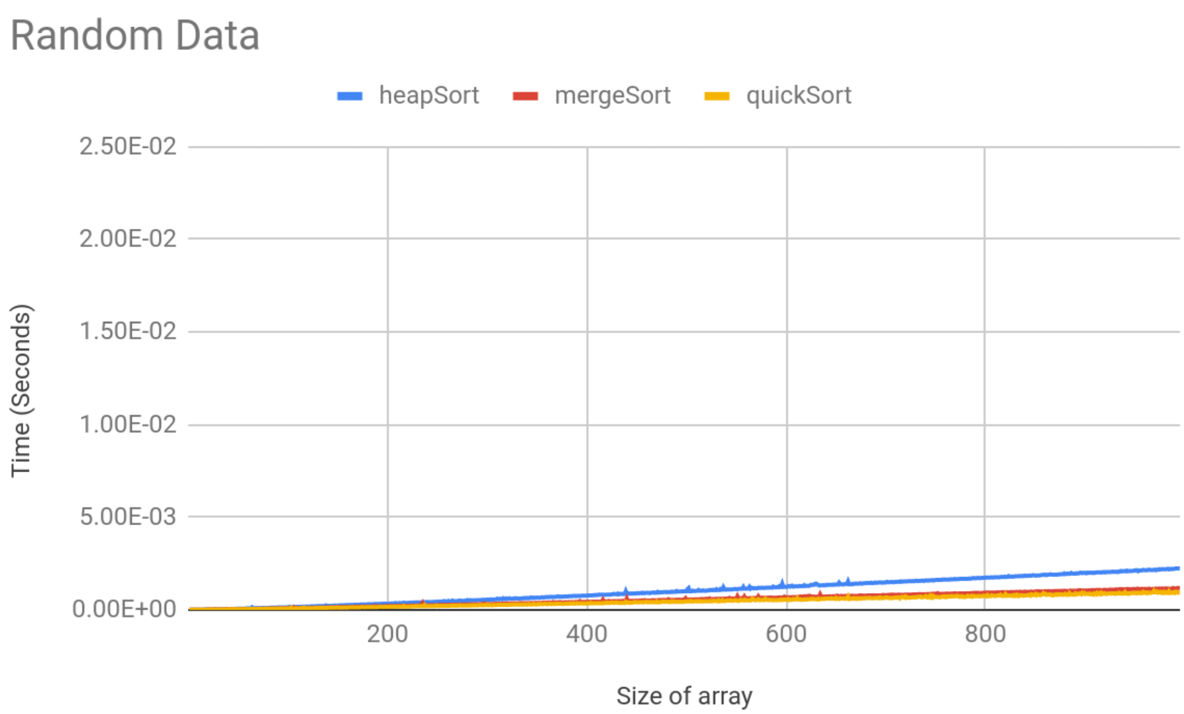
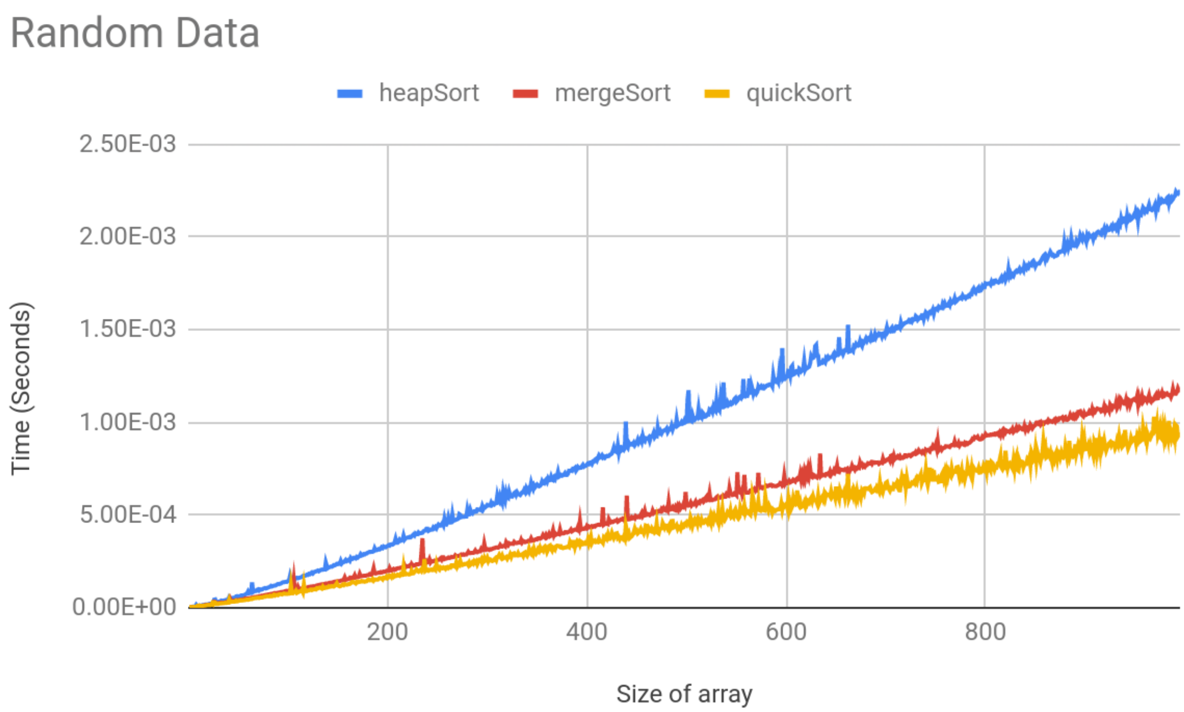
Jonathan Koning
koningj@oregonstate.edu

HW3







The only difference between the two Random Data graphs is the scale. Both are representing the same set of data.

Consider two *sorted* arrays *A* of size *m* and *B* of size *n.*
a) Design an efficient (O(log m + log n)) Divide-and-Conquer algorithm to find the k'th element in the merged array. To be efficient, you should do this without actually merging the two arrays.

Def kelement(A, B, k):

1) Check if the lengths of either of the arrays is 1.
   a) If so, determine the larger array. Let's assume A is the larger array.
   b) If k == 1, return min(A[0], B[0]).
   c) If k == 2, return max(A[0], B[0]).
   d) Else if A[k-1] < B[0], return A[k-1]
   e) Else return max(A[k-2], B[0])
2) Find mid indices, mid1 and mid2, of arrays A and B respectively
3) If mid1 + mid2 + 1 < k, we know that we will be dropping the first half of one of the arrays
   a) If A[mid1] < B[mid2], return kelement(A[mid1+1:], B, k-mid1-1). We drop the first half of A because it has the most small numbers before the kth element
   b) Else, return kelement(A, B[mid2+1:], k-mid2-1) for the same reasoning
4) If mid1+mid2+1>k, we know that we will be dropping the second half of one of the arrays
   a) If A[mid1] < B[mid2], return kelement(A, B[:mid2+1], k). We drop the second half of B because it has the most large numbers after the kth element

b) Else return kelement(A[:mid1+1], B, k). We drop the second half of A for the same reasoning

(b) Prove that the time complexity is *O(log n + log m).*

The algorithm above is O(log n + log m) because with each iteration, one of the arrays is divided in half. We stop dividing in half once one of the arrays is down to size 1. The maximum amount of times we can divide each array is log(size of array). So the worst case scenario is dividing both arrays until they are both of size 1. Because the size of one array is independent of the other, we must add the complexities together making the total complexity O(log n + log m)