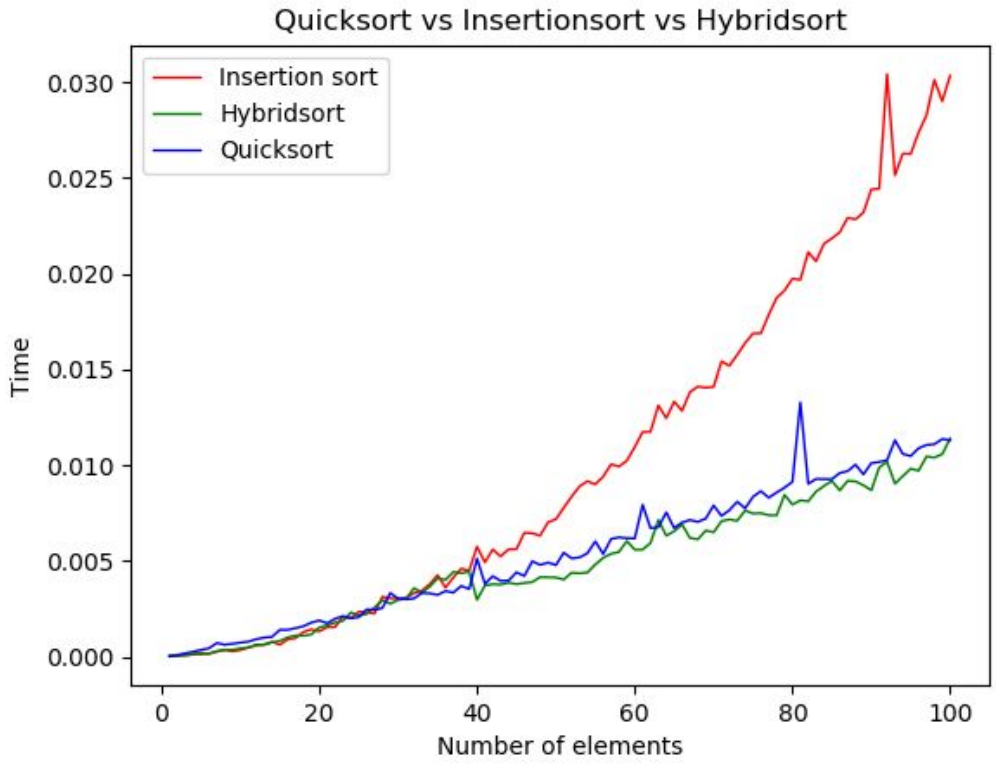
Question 2: Hybrid sorting

- *Hypothesis*:
  We hypothesize that the value for which we call insertion sort will be the same value for which quicksort and insertion sort intersect.
- *Methods*:
  We found that quicksort and insertion sort intersected at about 40 elements. Therefore we ran quicksort and insertion sort on lists that have between 35 and 45 elements 100,000 times and found the number of elements which insertion sort took longer than quicksort. We then averaged all of the values found on the 100,000 iterations. The value that resulted from 100,00 iterations was 40.4. Naturally, we rounded down to 40 for our hybrid sort.
- *Results*:
  We found that the value of the crossover produced the best time because it ran the faster sort for the given amount of elements.



  This is graph of al 3 sorts for k value of 40.
- *Discussion*:
  As a unit, we expected the graph to be equivalent to insertion sort below 40 elements and be equivalent to quicksort greater than 40 elements. The hybrid sort uses the faster section from each sort to create an optimized sort.
- *Conclusions*:
  With the insertion sort and quicksort we used, we found that insertion sort was faster for lists of size < 40. While quicksort was faster for lists of size > 40. At size 40, insertion and

quicksort have negligible time differences. Therefore we made a hybrid sort which calls insertion sort when the list is size < 40 and uses quicksort for lists of size > 40.