Project 1

Algorithm:

The idea is base on the the partition function, which is part of the quick sort.

(i) Initially, l = 1, r = 10000.

(ii) For the array index from l to r, it evenly chooses $(r-l)/769+2$ numbers and selects the biggest number from them. Use this number as the pivot for comparing.

(iii) Similar with the partition function in quick sort, it can find out the index of the pivot in the array, and put numbers smaller than it on the right side and numbers bigger on the left side. It takes O(n) time.

(iiii) For the index i return from the partition function,

If i > 40 then all the k numbers must on the left side, then we recursively run step (ii) from l to index.

If i <= 40 then index from l to i must be part of the k numbers, we put numbers index from l to i into Best array then recursively run step (ii) from index to r.

(iiiii) The recursive process ends when l > r. Then it uses merge sort to make the Best array in descending order.

1. theoretical WC

The theoretical wc occurs when everytime the partition function runs, it chooses the smallest $(r-l)/769+2$ numbers.

$\frac{10000}{2+\frac{10000}{796}}=15$, assume everytime we select 15 numbers, the comparison operation occurs $\frac{(10000+40)*664}{15}=444437$

And the worst comparison in merge sort is $2*40*log40=400$

So the total comparison will be 444837

2. theoretical expected WC

expected WC = $\sum_{k=10200}^{444837}(\frac{15!}{10000^{15}})^{664}*(1-(\frac{15!}{10000^{15}})^{664})*k$

3. theoretical AVG

In average case, we select 769 biggest numbers at first. The comparison operation occurs $9999+768+769=11536$

And the average comparison in merge sort is $40 * log40 = 200$

So the total comparison will be 11736

4. observed WC and AVG and explain inconsistencies

observed WC: 32133

observed AVG: 11719.47