Assignment 2

Name:Tian Zhou

NETID:tz164

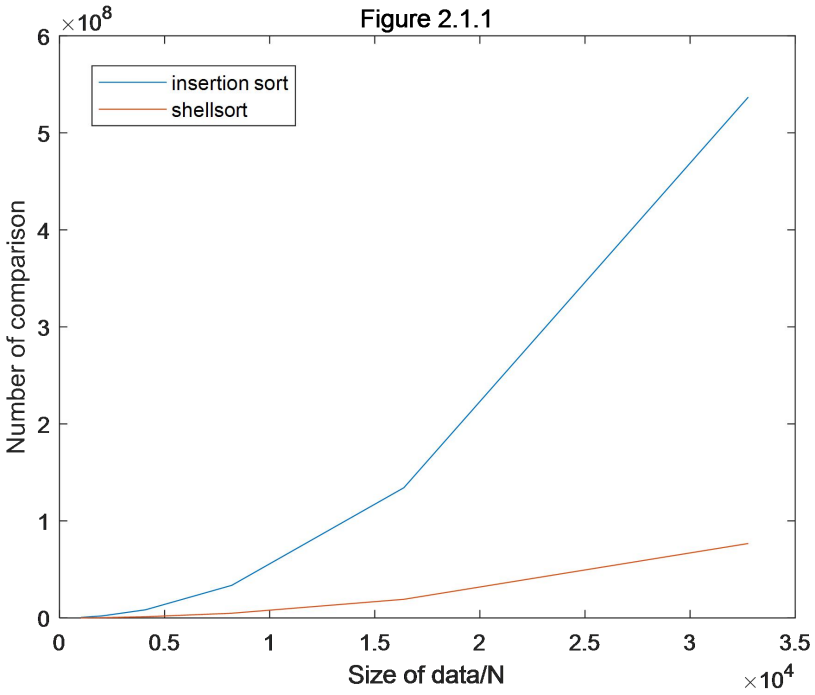
1.

When sorting the reversed order, I use the different size of data to compare the efficient of Shellsort and insertion sort.

I count the number of comparisons made in the sorting the data for insertion sort phase and shell sort phase.

	1024	2048	4096	8192	16384	32768
insertion	523776	2096128	8386560	33550336	134209536	536854528
shellsort	75555	299741	1199835	4798755	19175133	76707547

It's obvious that the number of comparisons for shellsort is much smaller than for insertion sort.



Explain: When we use the shellsort, the data which is in the wrong position could move more than one step for each movement. In that case, when we using the insertion sort for each part of the array, an element could easily reach his proper position.(fewer comparisons)

2.

size/N	distance	merge/N	NLogN	quadratic/N	N*N/2
1024	264541	10240	10240	524800	524288
2048	1027236	22528	22528	2098176	2097152
4096	4183804	49152	49152	8390656	8388608
8192	16928767	106496	106496	33558528	33554432
16384	66641183	229376	229376	134225920	134217728
32768	267933908	491520	491520	536887296	536870912

Quadratic algorithm

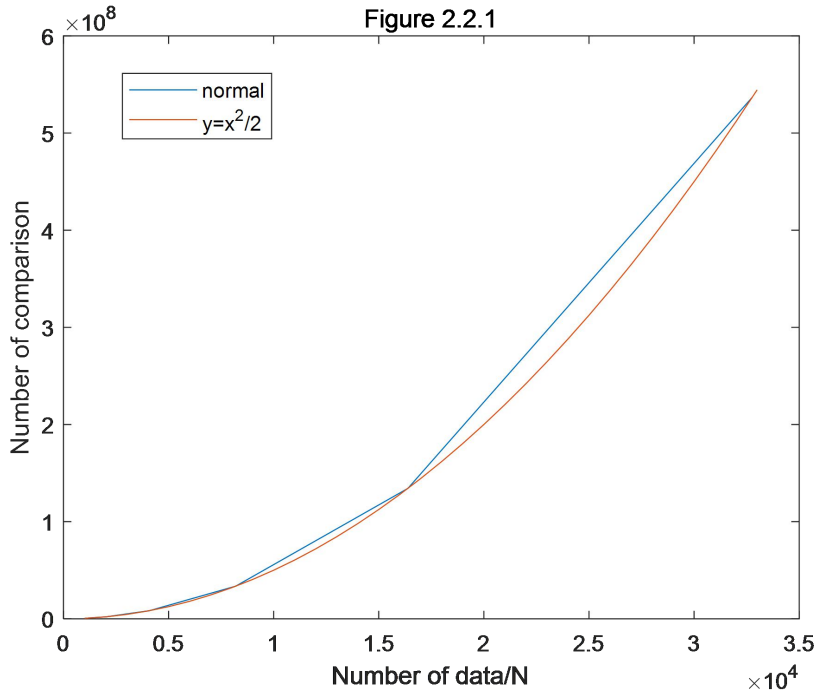
First of all, we index the array of data0, and record the position of each number in the array. Then, traverse the array of data1 and find out the position. Finally, find out the reverse pair of the elements and count the number of comparison.

Linearithmic algorithm: merge sort

While computing the Kendall Tau distance, we use the mergesort to sort the array and count the steps of movement.

When we are going to merge (lo, mid) and (mid+1,hi). Whenever we pick an element J from (mid+1,hi), the distance of movement is the number of elements in (lo,mid) which is larger than the element J. The result of distance by using mergesort is in the table.

Then, I count the number of operation of mergesort and compare it with quadratic algorithm. We can see that mergesort is a linearithmic algorithm which is less than quadratic algorithm.



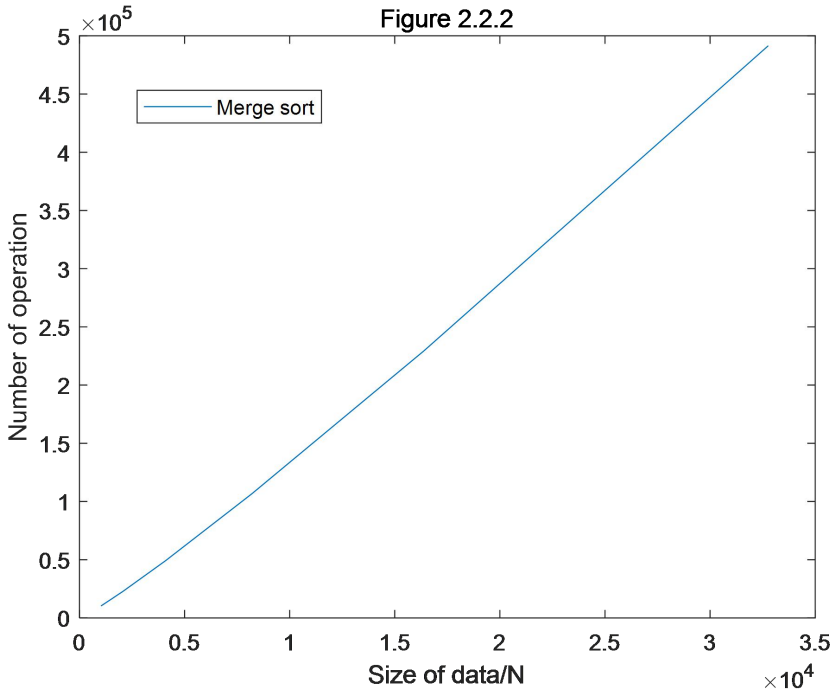


Figure 2.2.1 Normal method of counting distance (double loop)
Figure 2.2.2 Using Mergesort to count distance

Conclusion: The performance of mergesort method is more efficient than the other method.

3.

Because the array has many duplicate entries. I choose 3-way partitioning algorithm to sort this kind of data.

When sorting the reversed order, the number of operations is 12284. We can see that this algorithm is the most effective one.

Explain: While using 3-way partitioning algorithm, we scan i from left to right. When a[i] is equal to pivot value(used to be the first entry of array). We don't need to operate the array(exchange the values). Instead, we move the scan point i to the next point. The data set of 8192 entries only have 4 different values 1,11,111, and 1111. So, i think 3-way partitioning algorithm is the best algorithm.

4.

size	recursive	bottom-up
1024	10240	10240
2048	22528	22528
4096	49152	49152
8192	106496	106496
16384	229376	229376
32768	491520	491520

As we can see in the table above. The total number of comparisons to sort the data is exactly the $\mathbf{N\log_2 N}$ for both versions of mergesort when the data is of size **N**.

Explain: For each level of merge, it traversals the whole array **N**. An array could be divided as $\mathbf{\log_2 N}$ levels. So the total number of comparisons is $\mathbf{N\log_2 N}$.

5.

The table below contains the number of operations for each size of data while using different ways of sort.

size	N=7 to cut-off	quicksort	mergesort
1024	4777	4998	10240
2048	10534	10934	22528
4096	22851	23786	49152
8192	49676	51344	106496
16384	106519	109992	229376
32768	230072	237086	491520

As we can see in the table above. The number of operation for quicksort is approximately $\mathbf{(N\log_2 N)/2}$ which is smaller than the number for mergesort $\mathbf{N\log_2 N}$. When the number of data set is smaller or equal to 7, we cut-off the quicksort program as the insertion sort. It's obvious that insertion sort is much more efficient than the quicksort when the size of data is small. On the table, the number of operation for "cut-off to insertion" quicksort is a little bit smaller than quicksort.

6.

1.Mergesort (bottom-up)
2.Quicksort (standard, no shuffle)
3.Knuth shuffle
4.Mergesort(top-down)
5.Insertion sort
6.Heapsort
7.Selection sort
8.Quicksort (3-way,no shuffle)