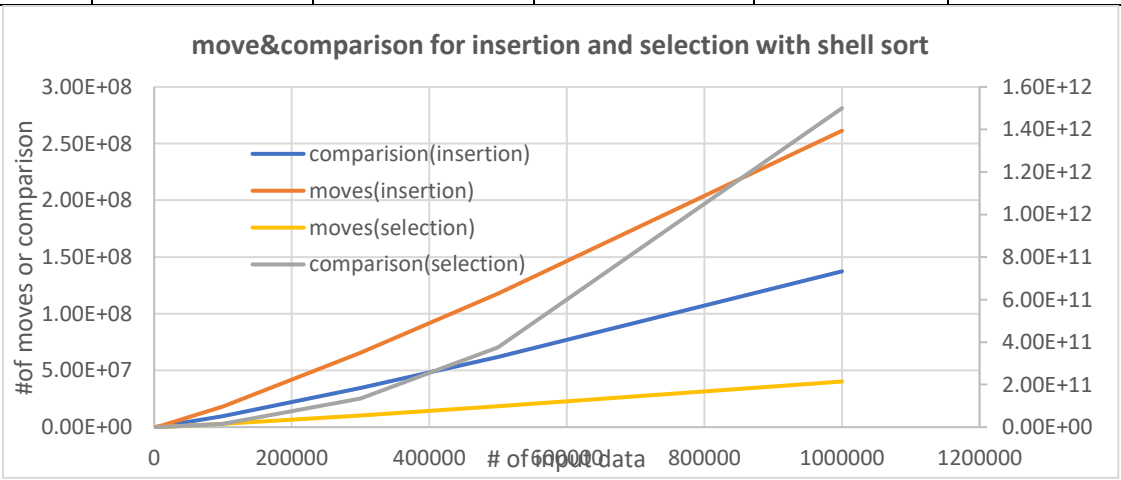
Project 1 Report Da Cheng id:0028653515

1.For my algorithm of the sequence, its time complexity is O((log(n))^2) and its space complexity is O(1).

2.My shellsort selection algorithm could run all the data sets below without any problem. When using shellsort with insertion or selection to sort data from small size to large size, the number of comparisons and moves are increased by an exponential relationship. Also, from the graph below, for the same number of elements to be sorted, the shellsort with insertion beats the shellsort with selection for the number of comparisons, and the shellsort with selection beats the shellsort with insertion for the number of moves. The reason is that selection has the least the number of moves. Also, the reason why my shellsort selection algorithm run very large input data sets inefficiently is that its big notation is O(n^2), which would grow extremely fast when input data set become very large. By contrast, the time complexity for insertion is O(n(log(n))^2), which cost very less comparison steps to run. One optimization method I tried is to obtain the index of maximum and minimum data and completing swap in one iteration simultaneously. Actually, this method only works with bubble sort. Then I tried to change the way to find subarrays and eliminated the repetitive comparisons steps by adding if structure. Finally, I successfully reduced the number of comparisons 100 times less for every data sets. In summary, the shellsort with insertion is more efficient.

# of elements	1000	10000	100000	300000	500000	1000000
shellsort with insertion	comparisons: 3.556800e+04 moves: 6.652300e+04	comparisons: 6.227570e+05 moves: 1.173468e+06	comparisons: 9.617893e+06 moves: 1.822330e+07	comparisons: 3.443823e+07 moves: 6.543018e+07	comparisons: 6.199436e+07 moves: 1.178865e+08	comparisons: 1.373963e+08 moves: 2.613834e+08
shellsort with selection	comparisons: 1.514965e+06 moves: 1.383900e+04	comparisons: 1.502731e+08 moves: 2.161380e+05	comparisons: 1.500429e+10 moves: 3.037446e+06	comparisons: 1.350156e+11 moves: 1.033886e+07	comparisons: 3.750280e+11 moves: 1.830660e+07	comparisons: 1.500062e+12 moves: 4.022734e+07



3.For the space complexity of my selection and insertion algorithm, they have the same big O notation which is O(n).

Reference

Wikipedia contributors. "Shellsort." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 4 Jun. 2018. Web. 26 Jun. 2018.