Student Number: 109550096   Name: Du Feng

Program assignment3 report

1. Simple depict about merge sort:

   At first, we have an original array with several numbers. Then we divide the original array into two new arrays (I called these two array as left and right in my code.). Continue this step of "dividing" with recursion until it reaches the end which means the size of the last new array is 1. After this, we start the merge with comparing number in each array until its size go back to the size of the original array.

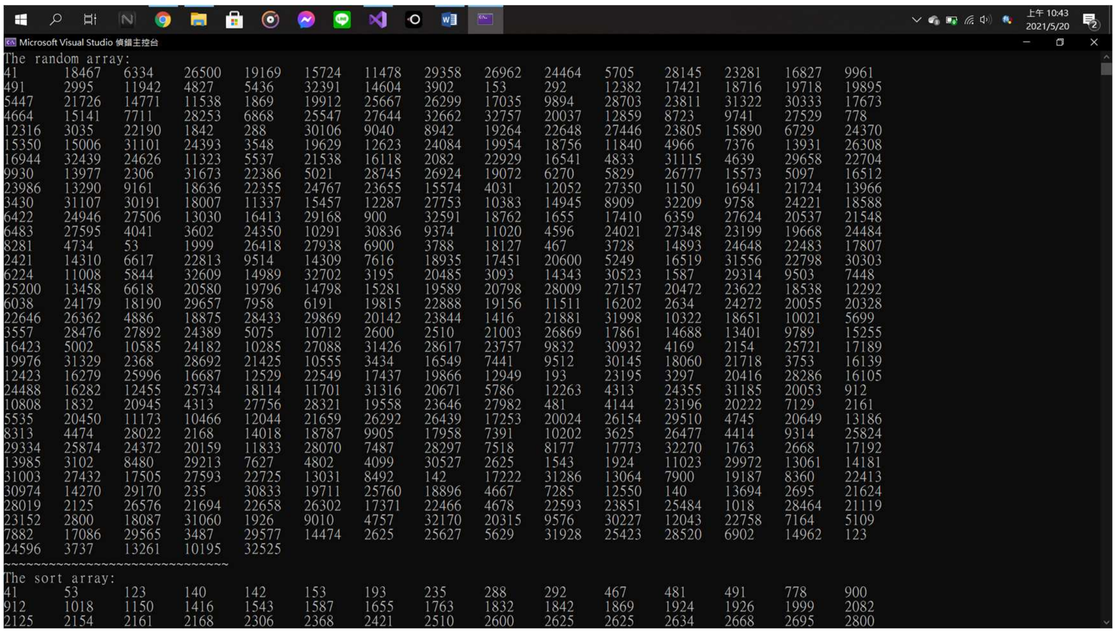
2. Explain its complexity and the reason:

   At first step, we use the loop "for" to divide the original array into two arrays (the first one goes to left array, the second one goes to right array, the third one goes to left array, the forth one goes to right array……). For n numbers, the complexity is 'n'. And if we want to divide this array 1/2 by 1/2 until the number of its size reaches 1. Its complexity is log(n). So, in the "divide" step its complexity is equal nlog(n).

   In the next step, we also use the loop "for" to compare the numbers in each array, and the complexity is n. Because there are n numbers, and we are merge it two by two. So its complexity is log(n). In this step, the complexity is nlog(n).

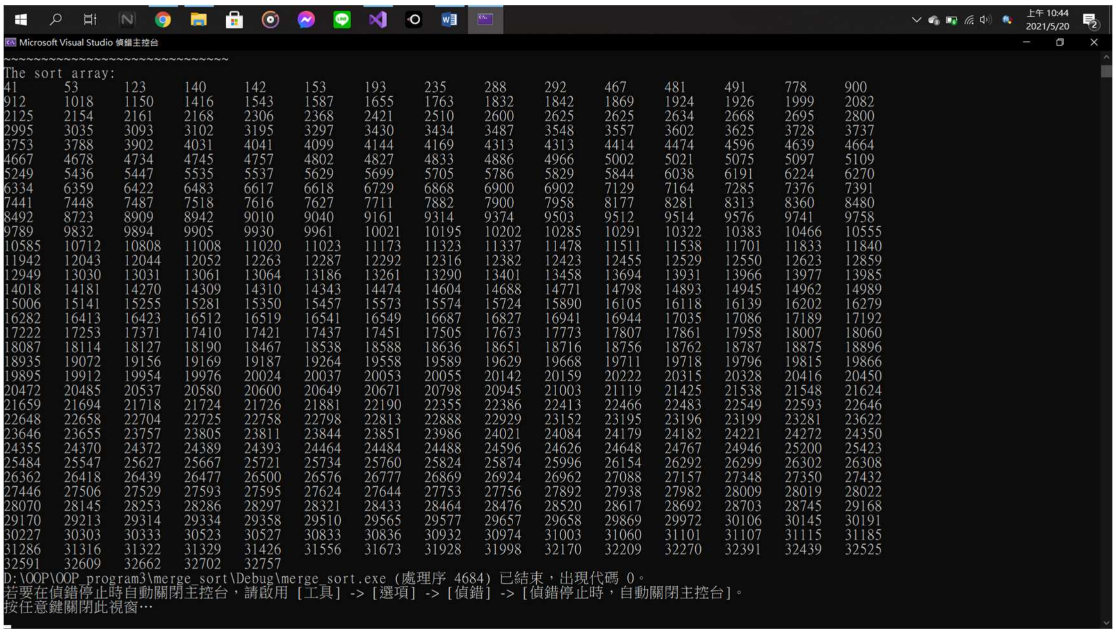
   In total, the complexity is nlog(n)+nlog(n)=2nlog(n). In the big O, the complexity will equal nlog(n).

3. Result-before: 500 random numbers for each line is 15 numbers



```
The random array:
41      18467   6334    26500   19169   15724   11478   29358   26962   24464   5705    28145   23281   16827   9961
491     2995    11942   4827    5436    32391   14604   3902    153     292     12382   17421   18716   19718   19895
5447    21726   14771   11538   1869    19912   25667   26299   17035   9894    28703   23811   31322   30333   17673
4664    15141   7711    28253   6868    25547   27644   32662   32757   20037   12859   8723    9741    27529   778
12316   3035    22190   1842    288     30106   9040    8942    19264   22648   27446   23805   15890   6729    24370
15350   15006   31101   24393   3548    19629   12623   24084   19954   18756   11840   4966    7376    13931   26308
16944   32439   24626   11323   5537    21538   16118   2082    22929   16541   4833    31115   4639    29658   22704
9930    13977   2306    31673   22386   5021    28745   26924   19072   6270    5829    26777   15573   5097    16512
23986   13290   9161    18636   22355   24767   23655   15574   4031    12052   27350   1150    16941   21724   13966
3430    31107   30191   18007   11337   15457   12287   27753   10383   14945   8909    32209   9758    24221   18588
6422    24946   27506   13030   16413   29168   900     32591   18762   1655    17410   6359    27624   20537   21548
6483    27595   4041    3602    24350   10291   30836   9374    11020   4596    24021   27348   23199   19668   24484
8281    4734    53      1999    26418   27938   6900    3788    18127   467     3728    14893   24648   22483   17807
2421    14310   6617    22813   9514    14309   7616    18935   17451   20600   5249    16519   31556   22798   30303
6224    11008   5844    32609   14989   32702   3195    20485   3093    14343   30523   1587    29314   9503    7448
25200   13458   6618    20580   19796   14798   15281   19589   20798   28009   27157   20472   23622   18538   12292
6038    24179   18190   29657   7958    6191    19815   22888   19156   11511   16202   2634    24272   20055   20328
22646   26362   4886    18875   28433   29869   20142   23844   1416    21881   31998   10322   18651   10021   5699
3557    28476   27892   24389   5075    10712   2600    2510    21003   26869   17861   14688   13401   9789    15255
16423   5002    10585   24182   10285   27088   31426   28617   23757   9832    30932   4169    2154    25721   17189
19976   31329   2368    28692   21425   10555   3434    16549   7441    9512    30145   18060   21718   3753    16139
12423   16279   25996   16687   12529   22549   17437   19866   12949   193     23195   3297    20416   28286   16105
24488   16282   12455   25734   18114   11701   31316   20671   5786    12263   4313    24355   31185   20053   912
10808   1832    20945   4313    27756   28321   19558   23646   27982   481     4144    23196   20222   7129    2161
5535    20450   11173   10466   12044   21659   26292   26439   17253   20024   26154   29510   4745    20649   13186
8313    4474    28022   2168    14018   18787   9905    17958   7391    10202   3625    26477   4414    9314    25824
29334   25874   24372   20159   11833   28070   7487    28297   7518    8177    17773   32270   1763    2668    17192
13985   3102    8480    29213   7627    4802    4099    30527   2625    1543    1924    11023   29972   13061   14181
31003   27432   17505   27593   22725   13031   8492    142     17222   31286   13064   7900    19187   8360    22413
30974   14270   29170   235     30833   19711   25760   18896   4667    7285    12550   140     13694   2695    21624
28019   2125    26576   21694   22658   26302   17371   22466   4678    22593   23851   25484   1018    28464   21119
23152   2800    18087   31060   1926    9010    4757    32170   20315   9576    30227   12043   22758   7164    5109
7882    17086   29565   3487    29577   14474   2625    25627   5629    31928   25423   28520   6902    14962   123
24596   3737    13261   10195   32525
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The sort array:
41      53      123     140     142     153     193     235     288     292     467     481     491     778     900
912     1018    1150    1416    1543    1587    1655    1763    1832    1842    1869    1924    1926    1999    2082
2125    2154    2161    2168    2306    2368    2421    2510    2600    2625    2625    2634    2668    2695    2800
```

Result-after: 500 random numbers after merge sorted for each line is 15 numbers:



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The sort array:
41      53      123     140     142     153     193     235     288     292     467     481     491     778     900
912     1018    1150    1416    1543    1587    1655    1763    1832    1842    1869    1924    1926    1999    2082
2125    2154    2161    2168    2306    2368    2421    2510    2600    2625    2625    2634    2668    2695    2800
2995    3035    3093    3102    3195    3297    3430    3434    3487    3548    3557    3602    3625    3728    3737
3753    3788    3902    4031    4041    4099    4144    4169    4313    4313    4414    4474    4596    4639    4664
4667    4678    4734    4745    4757    4802    4827    4833    4886    4966    5002    5021    5075    5097    5109
5249    5436    5447    5535    5537    5629    5699    5705    5786    5829    5844    6038    6191    6224    6270
6334    6359    6422    6483    6617    6618    6729    6868    6900    6902    7129    7164    7285    7376    7391
7441    7448    7487    7518    7616    7627    7711    7882    7900    7958    8177    8281    8313    8360    8480
8492    8723    8909    8942    9010    9040    9161    9314    9374    9503    9512    9514    9576    9741    9758
9789    9832    9894    9905    9930    9961    10021   10195   10202   10285   10291   10322   10383   10466   10555
10585   10712   10808   11008   11020   11023   11173   11323   11337   11478   11511   11538   11701   11833   11840
11942   12043   12044   12052   12263   12287   12292   12316   12382   12423   12455   12529   12550   12623   12859
12949   13030   13031   13061   13064   13186   13261   13290   13401   13458   13694   13931   13966   13977   13985
14018   14181   14270   14309   14310   14343   14474   14604   14688   14771   14798   14893   14945   14962   14989
15006   15141   15255   15281   15350   15457   15573   15574   15724   15890   16105   16118   16139   16202   16279
16282   16413   16423   16512   16519   16541   16549   16687   16827   16941   16944   17035   17086   17189   17192
17222   17253   17371   17410   17421   17437   17451   17505   17673   17773   17807   17861   17958   18007   18060
18087   18114   18127   18190   18467   18538   18588   18636   18651   18716   18756   18762   18787   18875   18896
18935   19072   19156   19169   19187   19264   19558   19589   19629   19668   19711   19718   19796   19815   19866
19895   19912   19954   19976   20024   20037   20053   20055   20142   20159   20222   20315   20328   20416   20450
20472   20485   20537   20580   20600   20649   20671   20798   20945   21003   21119   21425   21538   21548   21624
21659   21694   21718   21724   21726   21881   22190   22355   22386   22413   22466   22483   22549   22593   22646
22648   22658   22704   22725   22758   22798   22813   22888   22929   23152   23195   23196   23199   23281   23622
23646   23655   23757   23805   23811   23844   23851   23986   24021   24084   24179   24182   24221   24272   24350
24355   24370   24372   24389   24393   24464   24484   24488   24596   24626   24648   24767   24946   25200   25423
25484   25547   25627   25667   25721   25734   25760   25824   25874   25996   26154   26292   26299   26302   26308
26362   26418   26439   26477   26500   26576   26777   26869   26924   26962   27088   27157   27348   27350   27432
27446   27506   27529   27593   27595   27624   27644   27753   27756   27892   27938   27982   28009   28019   28022
28070   28145   28253   28286   28297   28321   28433   28464   28476   28520   28617   28692   28703   28745   29168
29170   29213   29314   29334   29358   29510   29565   29577   29657   29658   29869   29972   30106   30145   30191
30227   30303   30333   30523   30527   30833   30836   30932   30974   31003   31060   31101   31107   31115   31185
31286   31316   31322   31329   31426   31556   31673   31928   31998   32170   32209   32270   32391   32439   32525
32591   32609   32662   32702   32757
D:\OOP\OOP_program3\merge_sort\Debug\merge_sort.exe (處理序 4684) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗…
```

4. How to code in practical?
   For the main function, I set an array for 500 numbers in the beginning. And use a “while” loop to push 500 random numbers into the array.
   Then, call the function ***print_the_arr()*** to print the array.
   Next, call the function ***merge_arr()*** to do the main task.
   In this function, we will judge that if the size of the input array is 1. If it is true, return and end the function. Otherwise, create two arrays named ‘left’ and ‘right’. Then using a “for” loop to divide the original array into these two arrays.
   And continue to call ***merge_arr()*** with using the property of recursion.
   After finishing the step of “divide”, which means the original array has been divided into n arrays with one integer in it, call the function ***merge()***.
   In this function, we will have 3 arrays, one is the original array, another is the left array, the other is right array. The original array is the array which is divided into the left array and the right array. Then compare the numbers in the left array and the right array, respectively. For which number is smaller than another, push this number back to the original array. Continuing this step until we compare all the “left” and “right” arrays, and pushing all the integer back to the original array.
   In the end, call the function ***print_the_arr()*** to print the sorted array in the main function.