Homework Assignment #5 (Due Date: April 15)

Do the following problems:

(1) For the list shown below, show each of the following sorting algorithm as shown in the lecture notes:
   64, 32, 79, 83, 67, 46, 96, 55, 68, 12.

   (a) Heapsort (page 5-3).
   (b) Merge-sort (page 5-5).
   (c) Quick-sort (page 5-9). Use the rightmost element as the pivot in each subproblem.

(2) For the list given below, demonstrate external sort (as pages 5-18 to 5-20 in the lecture notes), assuming $M = 3$:
   6, 1, 20, 4, 9, 18, 12, 3, 10, 22, 5, 17, 2, 11, 5, 12, 8, 15, 7.

(3) Given the array shown in Fig. 1 representing disjoint sets, draw the associated trees.

\[
\begin{array}{cccccccc}
-1 & 0 & 0 & -1 & 3 & -1 & -1 & 6 & 7 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}
\]

   Figure 1: Disjoint sets

(4) Assuming that initially there are 13 sets, each has a unique value from 0 to 12, perform the following unions using union-by-size, show the result of each union. When the sizes are the same, make the second tree be the subtree of the first tree. Note that a find returns a root, and a union will union two roots.
union(find(11), find(12))
union(find(8), find(9))
union(find(8), find(10))
union(find(12), find(10))
union(find(4), find(5))
union(find(4), find(6))
union(find(4), find(7))
union(find(0), find(1))
union(find(1), find(2))

(5) Illustrate the array for the final forest of the previous problem. Note that the roots are not simply -1 when using union-by-size.

(6) Repeat problem (4) with path-compression also included.

(7) Given the disjoint set array shown in Fig. 2, show the array obtained after a find(10) is performed if path compression is used.

\[
\begin{array}{ccccccccccc}
\text{-1} & 0 & 0 & 2 & 2 & 1 & 1 & 5 & 5 & 8 & 9 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{array}
\]

Figure 2: Disjoint sets

(8) Illustrate the trees in the final forest of the previous problem.