CSE 130 Lab 7
Sort Design

Pseudocode:

```
GET file_name from user
OPEN file_name
READ line as a full line of text in JSON from file_name and add to file_list

SET i_pivot <- length of file_list – 1                                  A
SET i_check <- 0                                                        B
SET i_largest <- 0                                                      C

swap(arr, index1, index2)
        SET temp <- arr[index1]
        SET arr[index1] <- arr[index2]
        SET arr[index2] <- temp

WHILE i_check < i_pivot                                                 D
        IF file_list[i_check] >= file_list[i_largest]                   E
                SET i_largest <- i_check                                F

        IF i_check == i_pivot – 1                                       G
                IF file_list[i_largest] > file_list[i_pivot]            H
                        swap(file_list, i_largest, i_pivot)             I
                SET i_pivot <- i_pivot – 1                              J
                SET i_largest <- 0                                      K
                SET i_check <- 0                                        L

        SET i_check <- i_check + 1                                      M

PUT file_list on the screen                                             N
END
```

Algorithmic Efficiency:
O(n) Efficiency/ Linear Performance
Linear algorithms are characterized by a loop where every element in the collection is visited once. Note that sometimes a library or a feature of the programming language may obscure this loop, but the loop must still exist.
Loop. There must be a loop controlled by the input in some way. Note that this loop could be hidden in recursion or it could be in a function that our code calls. • Every element visited. In most cases, linear algorithms visit every element in the input buffer. However, there are exceptions. An algorithm that visits every other element would still be O(n) but the equation would be cost = ½n.

Program Trace:
Using the following dataset

```
{

  "array": [ "52", "26", "39", "15" ]

}
```

	i_pivot	i_check	i_largest
A	3	/	/
B	3	0	/
C	3	0	0
D	3	0	0
E	3	0	0
F	3	0	0
G	3	0	0
M	3	1	0
D	3	1	0
E	3	1	0
G	3	1	0
M	3	2	0
D	3	2	0
E	3	2	0
G	3	2	0
H	3	2	0
I	3	2	0
J	2	2	0
K	2	2	0
L	2	0	0
D	2	0	0
E	2	0	0
F	2	0	0
G	2	0	0
M	2	1	0
D	2	1	0
E	2	1	0
G	2	1	0
H	2	1	0
J	1	1	0
K	1	1	0
L	1	0	0
D	1	0	0
E	1	0	0
G	1	0	0
H	1	0	0
J	0	0	0
K	0	0	0
L	0	0	0
D	0	0	0