```
GET filename
READ file
data_json = json.loads(file)
data_list = data_json["array"]
num_list = parse int(data_list)

list_size = length(num_list)

FOR i_pivot from list_size - 1 down to 0
    i_largest = i_pivot

    FOR i_check from 0 up to i_pivot
        IF num_list[i_check] >= num_list[i_largest]
            i_largest = i_check

    IF num_list[i_largest] >= num_list[i_pivot]
        swap num_list[i_largest] with num_list[i_pivot]

PUT "Sorted list: ", num_list
```

The Algorithmic efficiency is O(n^2)
I can justify this by the tell this algorithm has.

1. It has two loops
2. The two loops are nested.
3. The two loops visit each element

Test Case:

```
{
  "array": [ "26", "6", "90", "55" ]
}
```

Sorted list: [6, 26, 55, 90]

Trace Table:

Line	1st loop Iteration	2nd loop Iteration	filename	data_list	num_list	list_size	i_pivot	i_largest	i_check index	i_check value
1	/	/	test.json	/	/	/	/	/	/	/

2	/	/	test.json	/	/	/	/	/	/	/
3	/	/	test.json	/	/	/	/	/	/	/
4	/	/	test.json	["26", "6", "90", "55"]	/	/	/	/	/	/
5	/	/	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	/	/	/	/	/
7	/	/	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	/	/	/	/
9	1	/	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	/	/	/
10	1	/	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	/	/
12	1	1	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	0	26
13	1	1	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	0	26
12	1	2	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	1	6
13	1	2	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	1	6
12	1	3	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	2	90
13	1	3	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	55	2	90
14	1	3	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	90	2	90
16	1	3	test.json	["26", "6", "90", "55"]	[26, 6, 90, 55]	4	3	90	2	90
17	1	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	3	90	2	90
9	2	0	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	90
12	2	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	0	26
13	2	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	0	26
12	2	2	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	1	6

13	2	2	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	1	6
12	3	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	55
13	3	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	55
14	3	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	55
16	3	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	55
17	3	3	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	2	55	2	55
9	3	0	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	1	6	2	55
12	3	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	1	6	0	26
13	3	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	1	6	0	26
14	3	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	1	26	0	26
16	3	1	test.json	["26", "6", "90", "55"]	[26, 6, 55, 90]	4	1	26	0	26
17	3	1	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	1	26	0	26
9	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	26	0	26
10	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	26
12	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6
13	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6
14	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6
16	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6
17	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6
19	4	0	test.json	["26", "6", "90", "55"]	[6, 26, 55, 90]	4	0	6	0	6