* A is an ordered integer array with 10 elements from small to large

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Shell sort

n = 10

gap = n/2 = 5

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Check all time first element less than second element.

5 comparison , 0 displacement

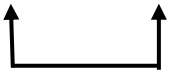
Gap size = gap size / 2.2 = 2

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19



Check all time first element less than second element.

8 comparison , 0 displacement

Gap equal to 2 so assign 1 to gap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Check all element 1 by 1 one and finish.

10 comparison , 0 displacement

Total : 23 comparison , 0 displacement

Merge Sort

No change just control

n=0	n=1	n=2	n=3	n=4		n=5	n=6	n=7	n=8	n=9
1	3	5	7	9		11	13	15	17	19

Split array to into 5-elemets arrays

n=0	n=1	n=2	n=3	n=4
1	3	5	7	9

Split array to into 2 and 3-elemets arrays

n=0	n=1
1	3

Split array to into 1-elemets arrays

n=0	n=1
1	3

Merge two 1-elements into a 2 elements array

n=2	n=3	n=4
5	7	9

Split array to into 1 and 2-elemets arrays

n=3	n=4
7	9

Split and merge

n=0	n=1	
1	3	
n=2	n=3	n=4
5	7	9

Merge 2 and 3 elements array into 5 elements and the other side same

Total 10 comparison, 0 displacement

Heapsort

First make heap array: parent bigger than children.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
19	17	15	13	11	9	7	5	3	1

10 comparison

Than remove first element and add to last and reheap.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	17	15	13	11	9	7	5	3	19

3 comparison 4 displacement

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
17	13	15	5	11	9	7	1	3	19

Than remove first element and add to last element in heap (changed element not in heap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
3	13	15	5	11	9	7	1	17	19

2 comparison 3 displacement

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
15	13	9	5	11	3	7	1	17	19

Than remove first element and add to last element in heap (changed element not in heap)

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
13	11	9	5	1	3	7	15	17	19

2 comparison 3 displacement

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Total: 20 displacement, 25 comparison

Quicksort

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Choose first element a pivot

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Compare all element and add array. Array is sorted so there is no displace element.

10 comparison, 0 displacement

*** B is an ordered integer array with 10 elements from large to small**

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
19	17	15	13	11	9	7	5	3	1

Shell sort

n = 10

gap = n/2 = 5

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
19	17	15	13	11	9	7	5	3	1

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
9	17	15	13	11	19	7	5	3	1

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
9	7	5	3	1	19	17	15	13	11

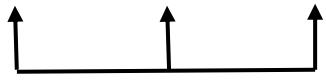
Check all time first element bigger than last and change it.

Gap size = gap size / 2.2 = 2

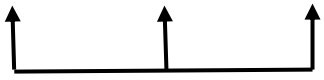
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
9	7	5	3	1	19	17	15	13	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
5	7	9	3	1	19	17	15	13	11

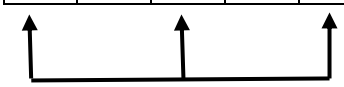
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
5	3	9	7	1	19	17	15	13	11



n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
5	3	1	7	9	19	17	15	13	11

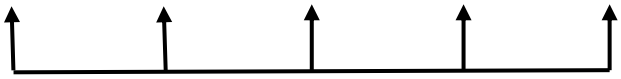


n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	19	17	15	13	11



:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19



Check all time first element less than second element.

And all item is sorted

Gap equal to 2 so assign 1 to gap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Check all element 1 by 1 one and finish.

Merge Sort

n=0	n=1	n=2	n=3	n=4		n=5	n=6	n=7	n=8	n=9
19	17	15	13	11		9	7	5	3	1

Split array to into 5-elemets arrays

n=0	n=1	n=2	n=3	n=4
19	17	15	13	11

Split array to into 2 and 3-elemets arrays

n=0	n=1
19	17

Split array to into 1-elemets arrays

n=0	n=1
17	19

Merge two 1-elements into a 2 elements array

n=2	n=3	n=4
15	13	11

Split array to into 1 and 2-elemets arrays

n=3	n=4
13	11

Split and merge

n=0	n=1
17	19

n=2	n=3	n=4
11	13	15

Merge 2 and 3 elements array into 5 elements and the other side same

n=0	n=1	n=2	n=3	n=4
11	13	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Heapsort

First make heap array: parent bigger than children.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
19	17	15	13	11	9	7	5	3	1

Than remove first element and add to last and reheap.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	17	15	13	11	9	7	5	3	19

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
17	13	15	5	11	9	7	1	3	19

Than remove first element and add to last element in heap (changed element not in heap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
3	13	15	5	11	9	7	1	17	19

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
15	13	9	5	11	3	7	1	17	19

Than remove first element and add to last element in heap (changed element not in heap)

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
13	11	9	5	1	3	7	15	17	19

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

Quicksort

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
19	17	15	13	11	9	7	5	3	1

Choose first element a pivot and compare all element.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
17	19	15	13	11	9	7	5	3	1

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
15	17	19	13	11	9	7	5	3	1

Add all element to first and it will be:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9
1	3	5	7	9	11	13	15	17	19

* C = {5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11}

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	13	9	1	7	6	8	1	15	4	11

Shell sort

n = 12

gap = n/2 = 6

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	13	9	1	7	6	8	1	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	13	9	1	7	6	8	1	15	4	11

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	1	9	1	7	6	8	13	15	4	11

Check all time first element bigger than last and change it.

Gap size = gap size / 2.2 = 2

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	5	9	1	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	1	9	1	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	5	9	1	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	9	5	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	9	5	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	9	5	7	6	8	13	15	4	11

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	7	5	9	6	8	13	15	4	11

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	7	4	8	5	9	6	11	13	15

Check all time first element less than second element.

And all item is sorted

Gap equal to 2 so assign 1 to gap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	7	4	8	5	9	6	11	13	15

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	2	1	7	4	8	5	9	6	11	13	15

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	1	2	4	5	6	7	8	9	11	13	15

Check all element 1 by 1 one and finish.

Merge Sort

n=0	n=1	n=2	n=3	n=4	n=5		n=6	n=7	n=8	n=9	n=10	n=11
5	2	13	9	1	7		6	8	1	15	4	11

Split array to into 6-elemets arrays

n=0	n=1	n=2	n=3	n=4	n=5
5	2	13	9	1	7

Split array to into 3-elemets arrays

n=0	n=1	n=2
5	2	13

Split array to into 1 and 2-elements arrays

n=0	n=1
5	2

Merge two 1-elements into a 2 elements array

n=0	n=1	n=2
2	5	13

Merge them

n=0	n=1	n=2
2	5	13

n=3	n=4	n=5
9	1	7

Split array to into 1 and 2-elemets arrays

n=4	n=5
1	7

Split and merge

n=3	n=4	n=5
9	1	7

n=3	n=4	n=5
1	7	9

n=0	n=1	n=2	n=3	n=4	n=5
2	5	13	1	7	9

Merge 3 elements array into 6 elements and the other side same

n=0	n=1	n=2	n=3	n=4	n=5
1	2	5	7	9	13

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	1	2	4	5	6	7	8	9	11	13	15

Heapsort

First make heap array: parent bigger than children.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
15	13	11	8	9	7	6	5	1	1	4	2

Than remove first element and add to last and reheap.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
2	13	11	8	9	7	6	5	1	1	4	15

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
13	9	11	8	4	7	6	5	1	1	2	15

Than remove first element and add to last element in heap (changed element not in heap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
2	9	11	8	4	7	6	5	1	1	13	15

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
11	9	7	8	4	2	6	5	1	1	13	15

Than remove first element and add to last element in heap (changed element not in heap)

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
9	8	7	5	4	2	6	1	1	11	13	15

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	1	2	4	5	6	7	8	9	11	13	15

Quicksort

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
5	2	13	9	1	7	6	8	1	15	4	11

Choose first element for pivot

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
2	5	13	9	1	7	6	8	1	15	4	11

Compare second element with first element(pivot) and it less than first change them:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
2	5	13	9	1	7	6	8	1	15	4	11

Compare third element with second element(pivot) and add bigger element to next to pivot

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
2	5	9	13	1	7	6	8	1	15	4	11

Compare third element with second element(pivot) and add bigger element to next to pivot

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
4	1	1	2	5	11	6	8	7	9	13	15

Fourth element is pivot and sort 2 array :

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
1	1	2	4	5	6	7	8	9	11	13	15

*** D = {'S', 'B', 'I', 'M', 'H', 'Q', 'C', 'L', 'R', 'E', 'P', 'K'}**

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	M	H	Q	C	L	R	E	P	K

Shell sort

n = 12

gap = n/2 = 6

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	M	H	Q	C	L	R	E	P	K

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
C	B	I	M	H	Q	S	L	R	E	P	K

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	E	H	K	C	L	R	M	P	Q

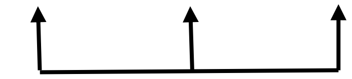
Compare each element and if first element less than change it.

Gap size = gap size / 2.2 = 2

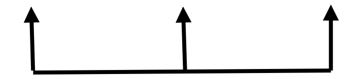
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	E	H	K	C	L	R	M	P	Q

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	B	S	E	H	K	C	L	R	M	P	Q

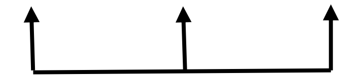
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	B	S	E	H	K	C	L	R	M	P	Q



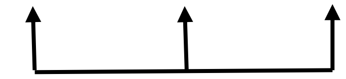
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	B	H	E	S	K	C	L	R	M	P	Q



n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
H	B	I	E	S	K	C	L	R	M	P	Q



n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
H	B	I	E	S	K	C	L	R	M	P	Q



n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
H	B	I	E	S	K	C	L	R	M	P	Q

:

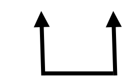
n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
C	B	H	E	I	K	P	L	R	M	S	Q

Compare elements with first element less than second element and change it if not.

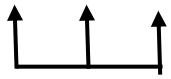
And all item is sorted

Gap equal to 2 so assign 1 to gap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
C	B	H	E	I	K	P	L	R	M	S	Q



n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	C	H	E	I	K	P	L	R	M	S	Q



:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	C	E	H	I	K	L	M	P	Q	R	S

Check all element 1 by 1 one and finish.

Merge Sort

n=0	n=1	n=2	n=3	n=4	n=5		n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	M	H	Q		C	L	R	E	P	K

Split array to into 6-elemets arrays

n=0	n=1	n=2	n=3	n=4	n=5
S	B	I	M	H	Q

Split array to into 3-elemets arrays

n=0	n=1	n=2
S	B	I

Split array to into 1 and 2-elements arrays

n=0	n=1
S	B

Merge two 1-elements into a 2 elements array

n=0	n=1	n=2
B	S	I

Merge them

n=0	n=1	n=2
B	I	S

n=3	n=4	n=5
M	H	Q

Split array to into 1 and 2-elemets arrays

n=4	n=5
H	Q

Split and merge

n=3	n=4	n=5
M	H	Q

n=3	n=4	n=5
H	M	Q

n=0	n=1	n=2	n=3	n=4	n=5
B	I	S	H	M	Q

Merge 3 elements array into 6 elements and the other side same

n=0	n=1	n=2	n=3	n=4	n=5

B	H	I	M	Q	S

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	C	E	H	I	K	L	M	P	Q	R	S

Heapsort

First make heap array: parent bigger than children.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	R	Q	M	P	K	C	B	L	H	E	I

Than remove first element and add to last and reheap.

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	R	Q	M	P	K	C	B	L	H	E	S

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
R	P	Q	M	H	K	C	B	L	I	E	S

Than remove first element and add to last element in heap (changed element not in heap

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
E	P	Q	M	H	K	C	B	L	I	R	S

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
Q	P	K	M	H	I	C	B	L	E	R	S

n remove first element and add to last element in heap (changed element not in heap)

Reheap:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
P	M	K	L	H	I	C	B	E	Q	R	S

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	C	E	H	I	K	L	M	P	Q	R	S

Quicksort

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
S	B	I	M	H	Q	C	L	R	E	P	K

Choose first element for pivot

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	S	I	M	H	Q	C	L	R	E	P	K

Compare second element with first element(pivot) and it less than first change them:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	B	S	M	H	Q	C	L	R	E	P	K

Compare third element with second element(pivot) and change them

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11

I	B	M	S	H	Q	C	L	R	E	P	K

Compare third element with second element(pivot) and change them

:

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
I	B	M	H	Q	C	L	R	E	P	K	S

Sort before pivot element

n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11
B	C	E	H	I	K	L	M	P	Q	R	S