CUBESORT

...

Introduction and History of Cubesort Algorithm

Cubesort is a parallel sorting algorithm which sort items using processor shuffle-exchange. It builds a self-balancing multi-dimensional array from the keys to be sorted. It's a comparison based sorting algorithm.

Cubesort is invented by **Cypher, Robert, Sanz and Jorge L.C** in **1992** and published in Journal of Algorithms(Volume 13, Issue 2, June 1992)

Running Time and Space Complexity

Cubesort best case running time : **O(n)**

Cubesort average case running time : **O(n log n)**

Cubesort worst case running time : **O(n log(n))**

Space Complexity: **O(n)**

Pseudo-code

```
cubesort(arr,n)
i = 0
for i to n-1
        for j = i+1 to arr.length
                a = pow(arr[i], 3)
                b = pow(arr[j], 3)

                if a > b
                temp = arr[i]
                arr[i] = arr[j]
                arr[j] = temp
```

Simulation of Cubesort

Before sorting Array is

4	-2	0	5	-9	3

i = 0

4	-2	0	5	-9	3

j = 1

4	-2	0	5	-9	3

After cubing 4 and -2 , a>b condition is true. the value will swap. J value will increase

-2	4	0	5	-9	3

Simulation Continues...

-2	4	0	5	-9	3

i = 0

-2	4	0	5	-9	3

j = 2

-2	4	0	5	-9	3

After cubing -2 and 0 , a>b condition is false. Value will not swap. J value will increase.

Simulation Continues...

-2	4	0	5	-9	3

i = 0

-2	4	0	5	-9	3

j = 3

-2	4	0	5	-9	3

After cubing -2 and 5 , a>b condition is false. Value will not swap. J value will increase.

Simulation Continues...

-2	4	0	5	-9	3

i = 0

-2	4	0	5	-9	3

j = 4

-2	4	0	5	-9	3

After cubing -2 and -9 , a>b condition is True. Value will swap. J value will increase.

Simulation Continues...

-9	4	0	5	-2	3

i = 0

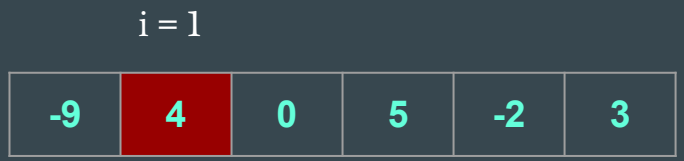
-9	4	0	5	-2	3

j = 5

-9	4	0	5	-2	3

After cubing -9 and 3 , a>b condition is False. Value will not swap. J value will end. Increase value of i

Simulation Continues...



After cubing 4 and 0 , a>b condition is True. Value will swap. J value will increase.

Simulation Continues...

-9	0	4	5	-2	3

i = 1

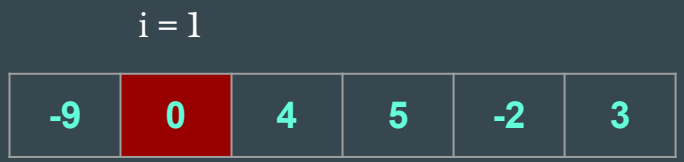
-9	0	4	5	-2	3

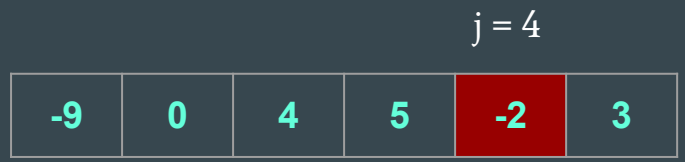
j = 3

-9	0	4	5	-2	3

After cubing 0 and 5 , a>b condition is False. Value will not swap. J value will increase.

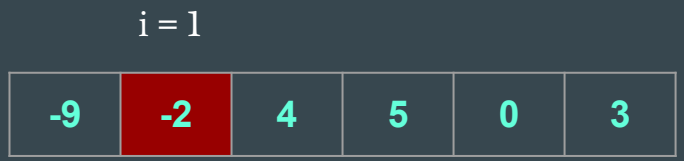
Simulation Continues...

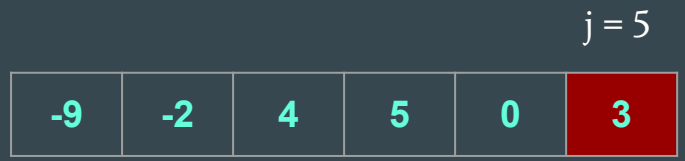




After cubing 0 and -2 , a>b condition is True. Value will swap. J value will increase.

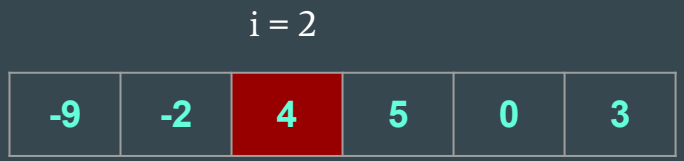
Simulation Continues...

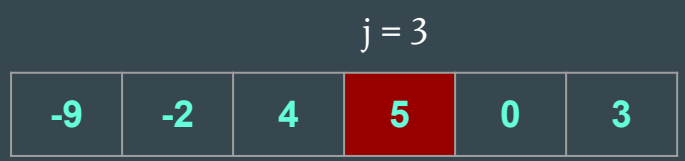




After cubing -2 and 3 , a>b condition is False. Value will not swap. J loop will end. Increase value of i

Simulation Continues...





After cubing 4 and 5 , a>b condition is False. Value will not swap. J value will increase

Simulation Continues...

-9	-2	4	5	0	3

i = 2

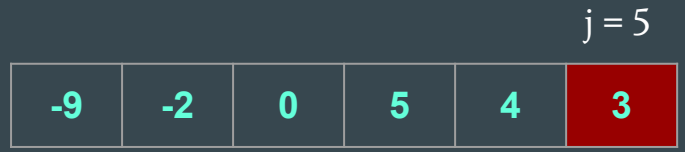
-9	-2	4	5	0	3

j = 4

-9	-2	4	5	0	3

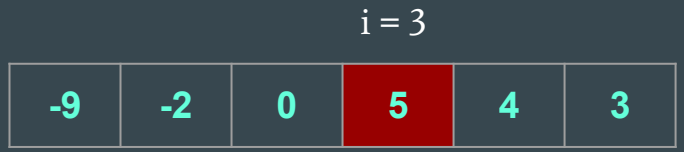
After cubing 4 and 0 , a>b condition is True. Value will swap. J value will increase

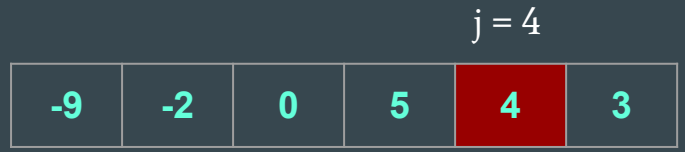
Simulation Continues...



After cubing 0 and 3 , a>b condition is False. Value will not swap. J loop will end. Increase value of i.

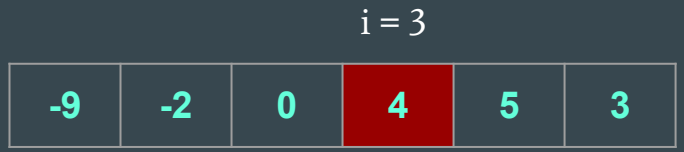
Simulation Continues...

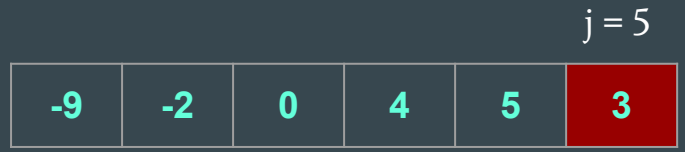




After cubing 5 and 4 , a>b condition is True. Value will swap. J value will increase

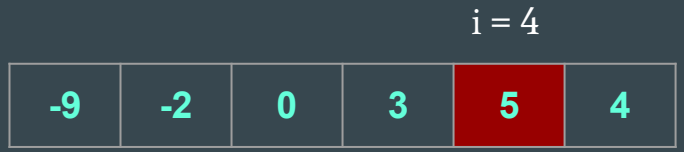
Simulation Continues...

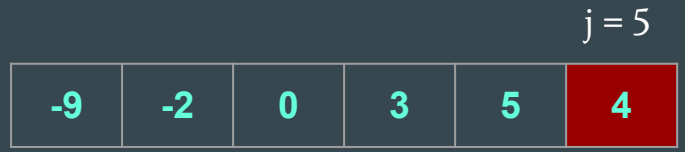




After cubing 4 and 3 , a>b condition is True. Value will swap. J loop will end. Increase value of i.

Simulation Continues...





After cubing 5 and 4 , a>b condition is True. Value will swap. J loop will end. Increase value of i.

Simulation Continues...

i = 5

j = i+1=5+1=6 invalid

i=5, j value is 6, j=6 is invalid. So i and j loop will be stop and remain array is the sorted array.

Attributes of Cubesort Algorithm

Stability: Cubesort is stable which means that the relative position of equal valued elements in the input and sorted array remains the same.

Out of place: Cubesort is an out of place algorithm because it requires O(n) extra space for sorting.

Adaptivity: Cubesort is adaptive which means that it can change behavior in running time based on available information.

Online or Offline: Cubesort is online which means it can input data while it is running.

Pros and Cons of Cubesort

Pros: Cube sort gives an additional speed advantage when sorting cubes compared to tree-based sorts. Also, cubesort is well-suited as an online or external sort. Insertions to the end are very fast memory operations.

Cons: If the data set is small, the memory overhead of cube sort becomes high.

Practical Uses of Cubesort

Cubesort can be used to sort N data items on-

- Hypercube
- Shuffle-exchange
- Cube-connected cycles computer

Best uses of Cubesort

Cubesort rapidly converts a 1-dimensional array than any other sorting algorithm

Presentation By

Farhan Zaman Ha-Mim

ID: 2022157

Md. Zahidul Islam

ID: 2022504

The End