3. Selection Sort

Theory -

→ Searching through the list of elements is the main algorithm of this sorting.
→ It swaps smallest element with the first element.
→ And repeats above process upto end of list.

Algorithm -

This algorithm sorts the array with N elements.

1. Repeat steps 2 and 3 for K = 1, 2, ..., N-1
2. Call MIN(A, K, N, LOC)
   => An array A is in memory. This procedure finds the location LOC of the smallest element among A[K], A[K+1], ... A[N]
   (i) Set MIN = A[K] and LOC = K [Initialize pointers]
   (ii) Repeat for J = K+1, K+2, ... N
   If MIN > A[J] then set MIN = A[J] and LOC = A[J], LOC = J
   [End of loop]
   (iii) Return
3. Interchange A[K] and A[LOC]
   Set Temp = A[K] A[K] = A[LOC] and A[LOC] = temp
   [End of step 1 loop]
4. Exit.

Process –

* Let us take an array of five elements A={9,36,8,48,7,56}
* In selection sort Method –

(i) First iteration –

{9,36,8,48,7,56}

After getting min – {7,36,8,48,9,56}

(ii) Second iteration –

{7,36,8,48,9,56}

After getting min – {7,8,36,48,9,56}

(iii) Third iteration –

{7,8,36,48,9,56}

After getting min – {7,8,9,48,36,56}

(iv) Fourth iteration – {7,8,9,48,36,56}

After getting min – {7,8,9,36,48,56}

(v) Fifth iteration –

{7,8,9,36,48,56}

After getting min = {7,8,9,36,48,56}

* After (n-1) iteration we will get sorted array.

Output -

[0, 1, 2, 3, 7, 13, 14]

Code -

```
a = [13,14,0,3,7,2,1]
n = len(a)
for i in range(n-1):
    min i = a[i]
    j = i+1
    while (j<n):
      if a[j] < min i:
          min i = a[j]
          value = j
      j = j+1
    a[i], a[value] = a[value], a[i]
print(a)
```