

I B.Tech. Computer Science & Business Systems

CSE209: Data Structures & Algorithms

Unit – IV: Sorting Selection Sort and Shell Sort

By
Dr. S. KAMAKSHI, AP-III / CSE
School of Computing



- Basic Principle
 - Select the ith minimum element and place it in the ith position
- Example
 - Given Array

25 15 10 45 22 3 79	16
---------------------	----

Pass 1: Minimum element 3, swap 25 and 3

25	15	10	45	22	3	79	82	16
		_		·				
3	15	10	45	22	25	79	82	16



- Pass 2: Next Minimum 10, swap 15 and 10

3	15	10	45	22	25	79	82	16
3	10	15	45	22	25	79	82	16

Pass 3: Next Minimum 15, no swap needed

3	10	15	45	22	25	79	82	16
3	10	15	45	22	25	79	82	16

- Pass 4: Next Minimum 16, swap 45 and 16

3	10	15	45	22	25	79	82	16
3	10	15	16	22	25	79	82	45



- Pass 5: Next Minimum 22, no swap needed

3	10	15	16	22	25	79	82	45
3	10	15	16	22	25	79	82	45

Pass 6: Next Minimum 25, no swap needed

3	10	15	16	22	25	79	82	45
3	10	15	16	22	25	79	82	45

- Pass 7: Next Minimum 45, swap 79 and 45

3	10	15	16	22	25	79	82	45
3	10	15	16	22	25	45	82	79
5	10	13	10		23	45	02	75



Pass 8: Next Minimum 79, swap 82 and 79

3	10	15	16	22	25	45	82	79
3	10	15	16	22	25	45	79	82

Final sorted array

3	10	15	16	22	25	45	79	82
---	----	----	----	----	----	----	----	----

Points to be noted

- When there are n elements, n-1 passes are required
- In ith pass over the array, select the minimum element from ith element to nth element
- Hence time complexity is O(n²)





Algorithm SELECTION_SORT(A,n)

```
1. for i = 1 to n - 1
2.
      minindex = i
3.
      for j = i + 1 to n
          if A[j] \leq A[minindex]
5.
               minindex = j
6.
      if minindex \neq i
7.
          temp = A[i]
          A[i] = A[minindex]
8.
          A[minindex] = temp
```



Basic Principle:

- Applying insertion sort on sub arrays of elements at distance d, for d = n/2, n/4, n/8, n/16, ..., 1
- When d=n/2, number of elements are lesser and the smaller number jumps d locations in single move



- Example
 - Given Array

1	2	3	4	5	6	7	8	9
25	15	10	45	22	3	79	82	16

- Pass 1: Split into sub arrays containing elements at distance 4

Sub arra 1	ays: 5	9	1	5	9
25	22	16	16	22	25
2	6		2	6	
15	3		3	15	
3	7		3	7	
10	79		10	79	

45 82

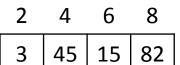
4	8
45	82
45	82



- After Pass1:
 - Array Contents are

 1	2	3	4	5	6	7	8	9
16	3	10	45	22	15	79	82	25

- Pass 2: Split into sub arrays containing elements at distance 2
 - Sub arrays:





1	3	5	/	9
10	16	22	25	79





- After Pass2:
 - Array Contents are

1	2	3	4	5	6	7	8	9
10	3	16	15	22	45	25	82	79

Pass 3: Whole array (Elements at distance 1)

1	2	3	4	5	6	7	8	9
3	10	15	16	22	25	45	79	82

- Points to be noted:
 - number of swaps are reduced drastically than normal insertion sort



Algorithm SHELL_SORT(A,n)

```
1. sv = n/2
2. while sv > 1
3.
     for k = 1 to sv
         for i = sv + k to n step sv
5.
            key = A[i]
6.
            j = i - sv
7.
            while j > 0 and A[j] \ge \text{key}
                  A[j + sv] = A[j]
8.
9.
                  j = j - sv
            A[j + sv] = key
10.
11.
    sv = sv/2
```