SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

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

Selection Sort

- Basic Principle
  - Select the i^{th} minimum element and place it in the i^{th} position
- Example
  - Given Array

25	15	10	45	22	3	79	82	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

Selection Sort

– 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

Selection Sort

– 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

Selection Sort

- 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 i^{th} pass over the array, select the minimum element from i^{th} element to n^{th} element
  - Hence time complexity is $O(n^2)$

Selection Sort

Algorithm SELECTION_SORT(A,n)

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

Shell Sort

- 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

Shell Sort

- 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 arrays:

1	5	9
25	22	16

1	5	9
16	22	25

2	6
15	3

2	6
3	15

3	7
10	79

3	7
10	79

    - .

4	8
45	82

4	8
45	82

Shell Sort

- 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	7	9
16	10	22	79	25

1	3	5	7	9
10	16	22	25	79

2	4	6	8
3	45	15	82

2	4	6	8
3	15	45	82

    - .

Shell Sort

- 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

Shell Sort

Algorithm SHELL_SORT(A,n)

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