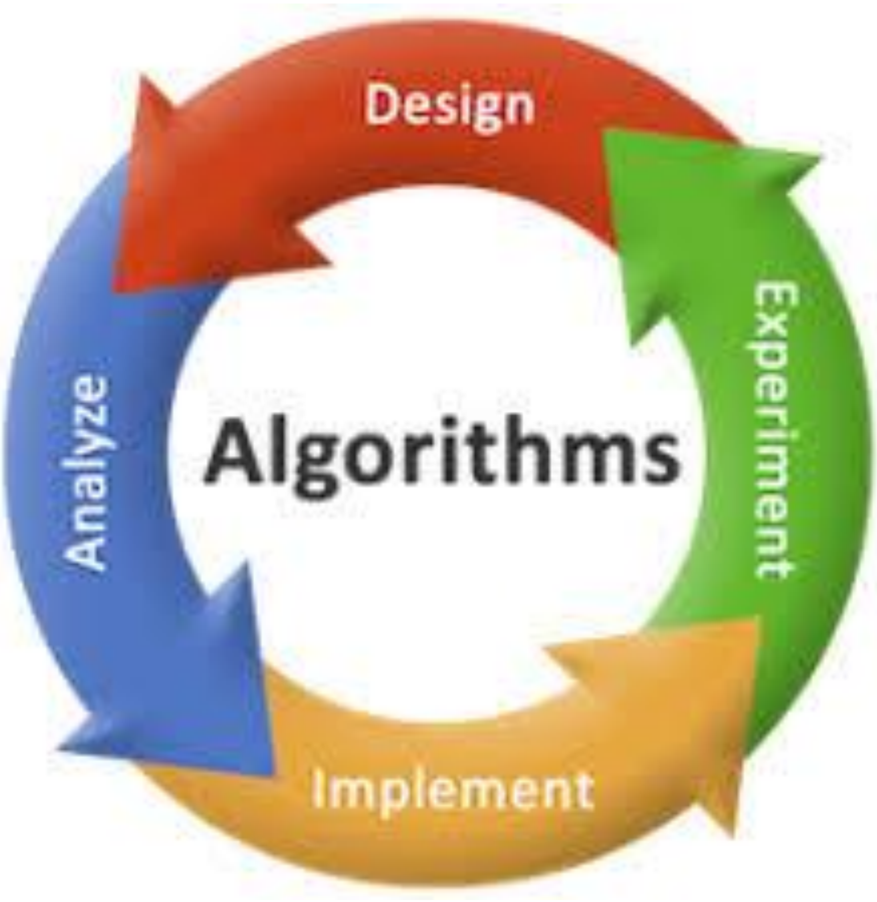

Design & Analysis of Algorithms

Lecture 10

Brute Force Approach for Problem Solving

Brute Force Approach

- ***Brute force*** is a straightforward approach to solving a problem, usually directly based on the problem statement and definitions of the concepts involved.

Brute Force Approach

Examples

- Linear/Sequential Search
- Sorting
  - Bubble
  - Insertion
  - Selection

Brute Force Approach

Sorting

- we consider the application of the brute-force approach to the problem of sorting: given a list of *n* orderable items (e.g., numbers, characters from some alphabet, character strings), rearrange them in ascending/descending order.

Brute Force Approach

Bubble Sorting

- **Brute-force** approach– to compare adjacent elements of the list and exchange them if they are out of order.
- By doing it repeatedly, we end up "bubbling up" the largest element to the last position on the list.
- The next pass bubbles up the second largest element, and so on, until after **$n-1$ passes** the list is sorted.

Brute Force Approach

Bubble Sorting

ALGORITHM $BubbleSort(A[0..n-1])$

//Sorts a given array by bubble sort
//Input: An array $A[0..n-1]$ of orderable elements
//Output: Array $A[0..n-1]$ sorted in nondecreasing order
for $i \leftarrow 0$ **to** $n-2$ **do**
  for $j \leftarrow 0$ **to** $n-2-i$ **do**
    if $A[j+1] < A[j]$ swap $A[j]$ and $A[j+1]$

Example: Sorting a list— 89, 45, 68, 90, 29, 34, 17

Brute Force Approach

Bubble Sorting

Example: Sorting a list— 89, 45, 68, 90, 29, 34, 17

Pass 1:

89	$\overset{?}{\leftrightarrow}$	45		68		90		29		34		17
45		89	$\overset{?}{\leftrightarrow}$	68		90		29		34		17
45		68		89	$\overset{?}{\leftrightarrow}$	90	$\overset{?}{\leftrightarrow}$	29		34		17
45		68		89		29		90	$\overset{?}{\leftrightarrow}$	34		17
45		68		89		29		34		90	$\overset{?}{\leftrightarrow}$	17
45		68		89		29		34		17	$\vert$	$\boxed{90}$

Brute Force Approach

Bubble Sorting

Example: Sorting a list – 89, 45, 68, 90, 29, 34, 17

Pass 2:

45	$\overset{?}{\leftrightarrow}$	68	$\overset{?}{\leftrightarrow}$	89	$\overset{?}{\leftrightarrow}$	29		34		17	|	90
45		68		29		89	$\overset{?}{\leftrightarrow}$	34		17	|	90
45		68		29		34		89	$\overset{?}{\leftrightarrow}$	17	|	90
45		68		29		34		17	|	$\boxed{89}$		$\boxed{90}$

etc.

Brute Force Approach

Bubble Sorting

Time Complexity

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=0}^{n-2-i} 1 = \sum_{i=0}^{n-2} [(n-2-i) - 0 + 1]$$

$$= \sum_{i=0}^{n-2} (n-1-i) = \frac{(n-1)n}{2} \in \Theta(n^2).$$

Brute Force Approach

Selection Sorting

- **Brute-force** approach– by scanning the entire given list to find its smallest element and exchange it with the first element, putting the smallest element in its final position in the sorted list.
- Then we scan the list, starting with the second element, to find the smallest among the last n−1 elements and exchange it with the second element, putting the second smallest element in its final position.

Brute Force Approach

Selection Sorting

ALGORITHM $SelectionSort(A[0..n-1])$

//Sorts a given array by selection sort
//Input: An array $A[0..n-1]$ of orderable elements
//Output: Array $A[0..n-1]$ sorted in nondecreasing order
for $i \leftarrow 0$ **to** $n-2$ **do**
    $min \leftarrow i$
    for $j \leftarrow i+1$ **to** $n-1$ **do**
        if $A[j] < A[min]$ $min \leftarrow j$
    swap $A[i]$ and $A[min]$

Example: Sorting a list— 89, 45, 68, 90, 29, 34, 17

Brute Force Approach

Selection Sorting

Example: Sorting a list— 89, 45, 68, 90, 29, 34, 17

| 89 45 68 90 29 34 **17**

17 | 45 68 90 **29** 34 89

17 29 | 68 90 45 **34** 89

17 29 34 | 90 **45** 68 89

17 29 34 45 | 90 **68** 89

17 29 34 45 68 | 90 **89**

17 29 34 45 68 89 | 90

Brute Force Approach

Selection Sorting

Time Complexity

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1-i) = \frac{(n-1)n}{2}.$$

$$= \Theta(n^2)$$

References

- **Chapter 3**: Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Pearson Education, Third Edition, 2017.

- **Chapter 2**: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, MIT Press/PHI Learning Private Limited, Third Edition, 2012.

Homework

Use brute-force approach to find time complexities of following algorithms:

- Linear/Sequential Search
- Insertion Sort
- Any other well-known sorting technique
- String Matching